

UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

IDENTIFICAÇÃO DE SISTEMAS ASSISTIDA POR COMPUTADOR

**DISSERTAÇÃO SUBMETIDA À UNIVERSIDADE FEDERAL
DE SANTA CATARINA PARA A OBTENÇÃO DE
GRAU DE MESTRE EM ENGENHARIA**

Sandro Carlos Lima

Florianópolis, dezembro - 1991

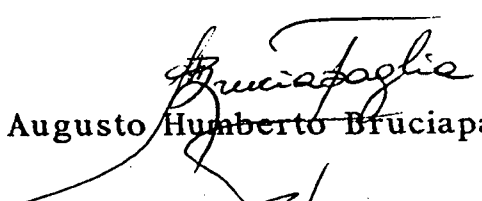
IDENTIFICAÇÃO DE SISTEMAS ASSISTIDA POR COMPUTADOR

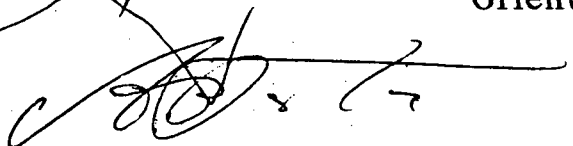
SANDRO CARLOS LIMA

ESTA DISSERTAÇÃO FOI JULGADA PARA A OBTENÇÃO DO TÍTULO DE

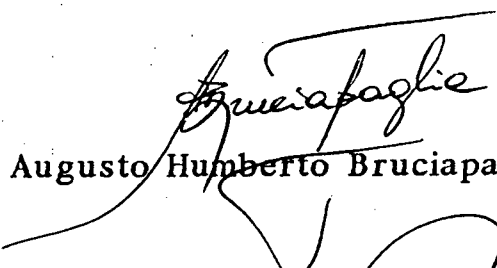
MESTRE EM ENGENHARIA

ESPECIALIDADE ENGENHARIA ELÉTRICA, ÁREA DE CONCENTRAÇÃO
SISTEMAS DE CONTROLE E AUTOMAÇÃO INDUSTRIAL, E APROVADA EM
SUA FORMA FINAL PELO CURSO DE PÓS-GRADUAÇÃO.


Prof. Augusto Humberto Bruciapaglia, Dr. Ing.
Orientador


Prof. João Pedro Assumpção Bastos, Dr. D'Etat
Coordenador do Curso de Pós-graduação em Engenharia Elétrica

BANCA EXAMINADORA:


Prof. Augusto Humberto Bruciapaglia, Dr. Ing.
Orientador


Prof. Rui Seara, Dr. Ing.
Co-Orientador


Prof. Julio Elias Normey Rico, M. Sc.


Prof. Enrique Luis Lima, D. Sc.

Ao meu filho, à minha esposa, e aos meus pais.

AGRADECIMENTOS

Aos professores Augusto Humberto Bruciapaglia e Rui Seara pela orientação e co-orientação deste trabalho. Aos demais professores, funcionários e colegas do LCMI, que direta ou indiretamente contribuíram para a sua conclusão, e pela amizade e convívio neste período. À CAPES pelo auxílio financeiro em forma de bolsa.

SUMÁRIO

RESUMO.....	IX
ABSTRACT.....	X
CAPÍTULO 1	
INTRODUÇÃO.....	1
CAPÍTULO 2	
O PROBLEMA DA IDENTIFICAÇÃO DE SISTEMAS	
REVISÃO BIBLIOGRÁFICA.....	4
2.1 INTRODUÇÃO.....	4
2.2 DEFINIÇÃO DO PROBLEMA DE IDENTIFICAÇÃO	4
2.2.1 Definição da Classe de Modelos.....	6
2.2.2 Aquisição de Dados	7
2.2.3 Escolha e Aplicação do Algoritmo	
de Identificação	8
2.2.4 Validação	9
2.3 TRATAMENTO DE SINAIS.....	10
2.4 CONCLUSÃO	12
CAPÍTULO 3	
ESPECIFICAÇÃO DO PROGRAMA ISAC	13
3.1 INTRODUÇÃO.....	13
3.2 REQUISITOS DO PROGRAMA	13
3.3 ESPECIFICAÇÃO DO PROGRAMA.....	14
3.3.1 Módulo de Gerenciamento	16

3.3.2 Módulo de Entradas..... 17

3.3.3 Módulo de Saídas 17

3.3.4 Módulo de Gerência de Arquivos..... 18

3.3.5 Módulo de Ferramentas 19

 3.3.5.1 Módulo de Geração de Sinais..... 19

 3.3.5.2 Módulo de Realização do
 Experimento 20

 3.3.5.3 Módulo de Simulação 20

 3.3.5.4 Módulo de Análise e Tratamento
 de Sinais..... 20

 3.3.5.5 Módulo de Identificação..... 21

 3.3.5.6 Módulo de Validação 22

3.4 CONCLUSÃO 22

CAPÍTULO 4.

IMPLEMENTAÇÃO DO ISAC..... 24

4.1 INTRODUÇÃO..... 24

4.2 O AMBIENTE DE EXECUÇÃO DO PROGRAMA..... 24

 4.2.1 Estrutura de Dados do Programa..... 24

 4.2.1.1 Diretório de Arquivos Virtuais 25

 4.2.1.2 Arquivos em Memória de Massa..... 26

 4.2.1.3 Gerenciamento da Estrutura de
 Dados..... 27

 4.2.2 A Interface com o Usuário..... 27

 4.2.3 Entrada de Dados e Apresentação de
 Resultados 28

4.3 GERAÇÃO DE SINAIS..... 31

 4.3.1 Sequência Binária Pseudo-Aleatória..... 32

4.4 REALIZAÇÃO DO EXPERIMENTO DE AQUISIÇÃO
 DE DADOS..... 33

4.5 SIMULAÇÃO DE SISTEMAS 35

 4.5.1 Simulação de Modelos Discretos 35

4.5.2 Edição de Arquivos e Discretização
de Modelos..... 37

4.6 ANÁLISE E TRATAMENTO DE SINAIS 37

4.6.1 Análise e Tratamento de Sinais no Tempo 37

4.6.2 Análise e Tratamento na frequência..... 40

4.7 A BASE DE MÉTODOS DE IDENTIFICAÇÃO 44

4.7.1 Algoritmos Baseados no Princípio
dos Mínimos Quadrados..... 44

4.7.1.1 Mínimos Quadrados Simples 45

4.7.1.2 Mínimos Quadrados Generalizados..... 46

4.7.1.3 Variáveis Instrumentais 47

4.7.1.4 Método da Matriz Estendida..... 48

4.7.2 Método da Correlação..... 48

4.7.3 Métodos Gráficos..... 49

4.7.3.1 Modelos de Segunda Ordem 50

4.7.3.2 Modelos de Alta Ordem Aperiódico 52

4.8 VALIDAÇÃO DE MODELOS..... 52

4.8.1 Algoritmo de Pré-determinação da
Ordem do Modelo 55

4.9 CONCLUSÃO 55

CAPÍTULO 5

RESULTADOS OBTIDOS COM A UTILIZAÇÃO DO PROGRAMA..... 57

5.1 INTRODUÇÃO..... 57

5.2 DESCRIÇÃO DOS EXPERIMENTOS 57

5.3 OBTENÇÕES DE INFORMAÇÕES INICIAIS SOBRE
O SISTEMA 58

5.3.1 Análise da Resposta ao Degrau 58

5.3.2 Pré-determinação da Ordem do Sistema..... 60

5.4 ANÁLISE DA INFLUÊNCIA DO SINAL DE EXCITAÇÃO 64

5.5 ANÁLISE DA INFLUÊNCIA DO PRÉ-TRATAMENTO
DE SINAIS 67

5.6 AVALIAÇÃO DO DESEMPENHO DOS ALGORITMOS.....	71
5.7 CONCLUSÃO.....	72
 CAPÍTULO 6	
CONCLUSÃO E PERSPECTIVAS	74
 BIBLIOGRAFIA.....	 78
 APÊNDICE - MANUAL DO USUÁRIO.....	 82

RESUMO

A presente dissertação aborda o desenvolvimento de um pacote computacional voltado à identificação de sistemas, o ISAC - Identificação de Sistemas Assistida por Computador.

O programa abrange sistemas monovariáveis, lineares ou linearizáveis em torno de um ponto de operação, englobando rotinas para a execução das principais etapas relacionadas ao procedimento de identificação, entre as quais pode-se citar: geração de sinais (1); simulação de sistemas (2); identificação (3) e validação (4) de modelos; análise e tratamento de sinais (5), e pré-determinação da ordem de modelos (6). No módulo de identificação o programa dispõe de algoritmos estatísticos (mínimos quadrados simples, mínimos quadrados generalizados, variáveis instrumentais, e método da matriz estendida), do método da correlação e de métodos gráficos para sistemas de primeira, segunda e alta ordem.

O desenvolvimento do ISAC baseia-se numa estrutura modular, sendo definida uma base de dados para interface entre os seus diversos módulos, permitindo-se a expansão do programa e a inclusão de novos algoritmos. Na interface com o usuário, foi desenvolvido um ambiente de alto nível, sendo considerada uma estrutura de menus para o controle do programa, a entrada de parâmetros e a apresentação de resultados em janelas, funções de auxílio ao usuário, e rotinas gráficas.

Resultados iniciais, obtidos por simulação, permitem concluir sobre o desempenho dos algoritmos e do ambiente de execução do programa.

ABSTRACT

In this work, the development of a computational package for systems identification - ISAC, portuguese acronym for Computer-Aided System Identification - is tackled.

This software covers monovariale linear (or linearized) systems bringing together many routines which allow one the execution of the main steps in the identification procedures, such as signal generation (1), system simulation (2), system identification (3), model validation (4), signal analysis and treatment (5), and pre-determination of the order of systems (6). In the identification module the software has statistical algorithms (such as least square, generalized least square, instrumental variables, and extended matrix method), correlation method, and graphical methods for first, second and high-order systems.

ISAC's development is based upon a modular structure consisting of a data base serving as an interface among its many modules. Such structure permits software extensions and inclusion of new algorithms. The user interface has been implemented through a high level environment. It consists of a menu-driven dialogue, a window-based parameter input and results displaying, on-line help, and graphical routines.

Initial results obtained by simulation allows one to conclude about the algorithms performance and the execution environment of the software.

CAPÍTULO 1

INTRODUÇÃO

No contexto das aplicações de sistemas de controle é indiscutível a importância da disponibilidade de modelos dinâmicos que representem adequadamente os processos sob estudo. Para a obtenção de tais modelos o engenheiro de controle pode recorrer às leis que regem os fenômenos envolvidos, para a determinação dos denominados "modelos de conhecimento". Entretanto, na maioria dos casos é necessário recorrer a aproximações, as vezes grosseiras, que deturpam os resultados obtidos por esta via.

Uma alternativa à sistemática acima descrita, consiste na obtenção dos "modelos de representação" dos sistemas, através da análise de seu comportamento entrada-saída, utilizando-se técnicas de identificação. É claro que entre as duas abordagens existem nuances.

Em função da variedade de sistemas, e das suas diferentes formas de representação, o procedimento de identificação de um processo real torna-se bastante complexo, sendo necessária, inicialmente, a definição da classe de modelos que se utilizará para a descrição de seu comportamento dinâmico. Além disto, para cada classe de modelo existem diversos métodos de identificação, desenvolvidos segundo diferentes abordagens, e que apresentam vantagens e desvantagens em relação à forma de cálculo, às características do experimento de aquisição de dados, ao armazenamento de dados, etc.

A utilização de um pacote computacional que reúna uma abrangente base de métodos e permita ao usuário a realização das principais etapas do procedimento de identificação, torna-se ideal. Desta forma, através de uma adequada interação homem-máquina, pode-se obter de uma maneira mais

eficiente o modelo do processo, não apenas pela utilização do método mais apropriado, como também pela comparação de modelos obtidos por diferentes métodos.

Neste contexto, foram desenvolvidos uma variedade de programas voltados à identificação de sistemas. Entre estes, pode-se citar "The System Identification Toolbox", uma extensão do MATLAB [Moler, 86], envolvendo uma coleção de ferramentas para todas as fases do procedimento de identificação, sendo considerada a identificação de modelos paramétricos e não paramétricos, a análise espectral de sinais, a simulação, validação, seleção da ordem de modelos, etc. Da mesma forma, diversos trabalhos tem sido realizados no Brasil na área de identificação de sistemas, entre os quais encontram-se: "Identificação em tempo real de sistemas lineares pelo método da correlação" [Cardoso F., 79]; "Pacote para identificação de processos biológicos" [Cardenas, 88]; "Ambiente para desenvolvimento e aplicação de algoritmos de identificação de sistemas" [Lima, 88]; e "Sistema de identificação, modelagem, análise e projeto de sistemas de controle para sistemas elétricos de potência" [Luz, 90]. Entretanto, na sua grande maioria, os programas desenvolvidos são voltados à identificação de sistemas específicos, ou consideram a utilização de um único método em particular.

Este trabalho trata do desenvolvimento de um pacote computacional, o ISAC - Identificação de Sistemas Assistida por Computador, destinado ao auxílio das principais etapas do procedimento de identificação de sistemas.

O ISAC, abrange o procedimento de identificação de uma forma geral, dispondo de uma base de métodos onde diferentes abordagens podem ser utilizadas para a obtenção do modelo do sistema. Além disto, são consideradas ferramentas para a geração de sinais, validação dos modelos identificados, e simulação de sistemas em ambiente acadêmico. O programa engloba ainda, ferramentas para a análise e pré-tratamento de sinais,

possibilitando-se a minimização da influência de ruídos e conseqüentemente um melhor desempenho dos algoritmos.

Desta forma, no Capítulo 2, discutir-se-a sumariamente o procedimento de identificação, enfocando as etapas envolvidas e os principais aspectos que devem ser considerados em cada uma delas.

No Capítulo 3, é apresentada a especificação do programa, envolvendo o seu escopo de abrangência e a descrição da sua estrutura modular, assim como as funções associadas a cada módulo.

A implementação dos módulos do programa, é discutida no Capítulo 4, sendo descritos os algoritmos de identificação escolhidos para a sua base de métodos. Além disto, faz-se um detalhamento das principais funções e algoritmos de todos os módulos do programa e do ambiente que foi desenvolvido para a execução destas funções.

Os resultados obtidos com a utilização do programa, são apresentados no Capítulo 5, sendo descritos os testes realizados e o desempenho dos algoritmos. Também neste capítulo, são verificadas as vantagens obtidas com a utilização de ferramentas para o pré-tratamento dos sinais de entrada e saída do sistema, quando utilizados em um procedimento de identificação.

Finalmente, o manual do usuário do programa, onde pode-se verificar, dentro de uma abordagem prática, as suas características operacionais, é apresentado no trabalho sob a forma de anexo.

CAPÍTULO 2

O PROBLEMA DA IDENTIFICAÇÃO DE SISTEMAS - REVISÃO BIBLIOGRÁFICA

2.1 INTRODUÇÃO

A teoria de identificação de sistemas teve, nas últimas décadas, um grande avanço com o desenvolvimento e generalização no uso de computadores. Os algoritmos de identificação foram aperfeiçoados e adaptados a casos particulares, sendo o procedimento de identificação, de uma maneira geral, sistematizado em etapas.

Neste capítulo, faremos uma descrição das principais etapas e aspectos práticos envolvidos no procedimento de identificação de sistemas e que foram considerados na especificação do programa ISAC - Identificação de Sistemas Assistida por Computador.

2.2 DEFINIÇÃO DO PROBLEMA DE IDENTIFICAÇÃO

Segundo Zadeh [62] o problema de identificação é formulado como sendo: "A determinação, com base na entrada e saída de um sistema, de um modelo, pertencente a uma classe específica, equivalente ao sistema sob teste". Nesta definição, destacam-se dois aspectos essenciais ao procedimento de identificação: a escolha de uma classe de modelos e o significado de equivalência.

A escolha de uma classe de modelos, ou caracterização, é uma etapa qualitativa, onde são levantadas as hipóteses sobre o sistema a ser identificado, e definida a classe do modelo que deverá representá-lo.

O significado de equivalência, dentro do contexto da formulação de Zadeh, é definido em termos de um critério a ser minimizado, através de

uma função envolvendo a saída do sistema " y " e a saída do modelo " y_m " [Åström, 71],

$$V = f(y, y_m).$$

A etapa quantitativa do procedimento de identificação, consiste na obtenção do modelo do sistema através de algoritmos baseados na minimização do critério de equivalência, como ilustra a Fig. 2.1. Mesmo quando o algoritmo de identificação, não utilize diretamente o critério de equivalência, a aplicação deste conceito implica na validação do modelo identificado em relação ao comportamento entrada-saída do sistema.

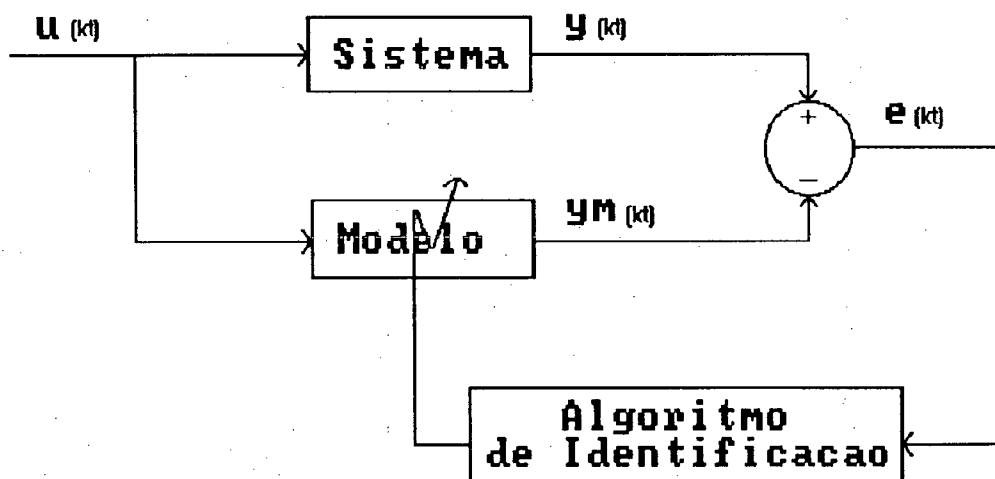


Fig. 2.1 - Aplicação do princípio de equivalência à Identificação de Sistemas

Em termos práticos, o procedimento de identificação envolve ainda, diversos aspectos relacionados à aquisição dos dados de entrada e saída do sistema. Desta forma, podemos dividir o procedimento de identificação de um dado processo em quatro etapas [Isermann, 80]:

- definição da classe de modelos;
- planejamento e execução do experimento de aquisição de dados;
- escolha e aplicação do algoritmo de identificação;
- validação do modelo identificado.

2.2.1 DEFINIÇÃO DA CLASSE DE MODELOS

No procedimento de identificação, o primeiro passo, consiste em definir a classe de modelos que será utilizada para a representação do sistema. Nesta etapa, com a análise das informações disponíveis sobre o comportamento do sistema, é possível opinar sobre características como a linearidade e invariância no tempo. Também, a utilização da abordagem "modelo de conhecimento", através do equacionamento das leis que governam o sistema, pode fornecer uma boa estimativa do tipo de sistema em questão, e até mesmo, em certos casos, da estrutura do modelo, reduzindo o problema de identificação a um simples caso de estimativa de parâmetros.

A questão primordial, que se pode levantar na etapa de caracterização é sobre a linearidade do sistema. Em geral, os processos industriais, quando não lineares, podem ser linearizados em torno de um ponto de operação.

Para a representação do comportamento dinâmico dos sistemas dispomos de diversos tipos de modelos que podem ser classificados segundo diferentes critérios: modelos contínuos e discretos; modelos a parâmetros concentrados e distribuídos; estocásticos e determinísticos; paramétricos e não paramétricos, etc. [Fasol, 80].

A escolha de um tipo específico de modelo, dentro da classe de modelos em que se enquadra o sistema, baseia-se, principalmente, na aplicação desejada. Por exemplo, considerando a utilização de um modelo no controle de processos, através de um computador digital, torna-se

desejável a representação do sistema através de modelos paramétricos e discretos.

2.2.2 AQUISIÇÃO DE DADOS

Consiste na obtenção dos dados de entrada e saída do sistema que serão utilizados nos algoritmos de identificação.

No procedimento de aquisição de dados, devem ser observados diversos aspectos que podem influenciar a identificação do sistema. Desta forma, além da definição do tipo de equipamento para a geração de sinal, leitura e armazenamento dos dados de saída, devem ser selecionados o período de amostragem e o sinal de excitação a ser utilizado.

Alguns métodos de identificação consideram a utilização de um tipo específico de sinal de entrada como, por exemplo, os algoritmos baseados na resposta ao degrau ou a seqüências binárias pseudo-aleatórias. Entretanto, de uma maneira geral, os sinais de excitação devem obedecer a condição de "excitação persistente", que implica em serem sinais suficientemente ricos para excitarem todos os modos de interesse do sistema [Åström, 71; Isermann, 80]. A escolha do sinal de excitação, bem como da sua amplitude, está diretamente relacionada às condições de operação do sistema. Desta forma, embora seja desejável a utilização de um sinal de "excitação persistente", com amplitude suficientemente grande, para minimizar a influência dos ruídos presentes, a escolha das características do sinal de excitação deve assegurar que o funcionamento normal do sistema não seja demasiadamente perturbado.

Além da escolha do sinal de excitação, a seleção do período de amostragem é um aspecto fundamental para o procedimento de identificação, atuando diretamente sobre a qualidade dos dados obtidos do sistema. Na seleção do período de amostragem deve-se considerar o tempo de amostragem da aplicação desejada, os problemas de precisão das operações aritméticas e os erros de quantização nos sinais de entrada e

saída e nos coeficientes do modelo. A experiência tem mostrado que a utilização da relação t_{95}/t_0 para valores entre 5 e 15, onde t_{95} é o tempo em que o sistema atinge 95% do regime permanente e t_0 é o período de amostragem, tem levado a resultados muito bons [Isermann, 80].

Quanto à forma de realização, o procedimento de identificação pode ser "on-line" ou "off-line". Na identificação "off-line" os dados são armazenados, podendo ou não serem processados, para uma posterior identificação, enquanto que no procedimento "on-line" os modelos são identificados por algoritmos recursivos à medida que os dados são adquiridos.

2.2.3 ESCOLHA E APLICAÇÃO DO ALGORITMO DE IDENTIFICAÇÃO

Nesta etapa são obtidos os modelos do sistema com a aplicação dos algoritmos de identificação. Para um mesmo tipo de modelo encontramos diversos algoritmos que podem ser classificados sob diferentes aspectos.

Inicialmente, quanto aos modelos a serem obtidos, podemos classificar os algoritmos em paramétricos, que resultam em modelos baseados em funções de transferência e equações de estado, e não paramétricos, resultando em modelos como a resposta ao impulso, resposta em frequência, etc. [Fasol, 80].

Considerando a forma de obtenção dos modelos destacam-se os algoritmos clássicos e estatísticos [Menendez, 79]. Os algoritmos clássicos baseiam-se sobre as propriedades gráficas da resposta temporal de um sistema (resposta ao degrau, resposta ao impulso, etc), sobre as propriedades de correlação entre os sinais de entrada e saída e sobre a resposta em frequência dos sistemas. Os algoritmos estatísticos (mínimos quadrados, variáveis instrumentais, etc.) baseiam-se na minimização de um critério de erro entre a saída do modelo e do sistema, para a estimação dos parâmetros do modelo.

Um dos principais aspectos a serem considerados na escolha e aplicação dos algoritmos de identificação é o conhecimento da estrutura do modelo a ser identificado. Quando o conhecimento prévio do sistema não é suficiente para a definição da estrutura do modelo, a aplicação dos algoritmos de identificação passa a ser um procedimento iterativo. Nestes casos, a utilização de modelos não paramétricos, pode fornecer uma boa estimativa da estrutura dos modelos paramétricos, reduzindo-se o número de iterações necessárias à identificação do sistema [Isermann, 80].

Quanto ao desempenho, verificamos que as características do sistema sob estudo e as condições em que se realizará o experimento (ruídos, sinal de excitação, período de amostragem) podem favorecer os resultados de alguns algoritmos [Isermann, 74; Saridis, 74]. Sendo assim, a utilização de mais de um método de identificação torna-se ideal, uma vez que com a validação de diferentes modelos pode-se escolher o mais adequado ao sistema em questão.

2.2.4 VALIDAÇÃO

A validação é a etapa em que todas as condições previamente definidas para a obtenção do modelo e o seu desempenho na representação do sistema, devem ser avaliadas.

Inicialmente, em relação à etapa de definição da classe do modelo, as características levantadas ou assumidas para o sistema, linearidade e ou invariância no tempo, podem ser avaliadas com a identificação e comparação de diversos modelos, obtidos a partir de sinais de diferentes amplitudes e utilizando-se séries de dados distintas.

Os métodos estatísticos, consideram em sua estrutura o critério de equivalência entre o sistema e o modelo a ser identificado. Entretanto, em função da presença de ruídos, da escolha do período de amostragem e do sinal de excitação, nada se pode afirmar sobre a qualidade dos dados utilizados, podendo o modelo obtido ser inadequado à representação do

sistema. Desta forma, seria desejável a verificação do comportamento entrada-saída do modelo, através da sua simulação, com diferentes sinais de excitação, e a comparação com a resposta real do sistema.

A validação também envolve a avaliação da estrutura escolhida para o modelo quando da aplicação do algoritmo de identificação. No caso linear monovariável, a validação da estrutura corresponde à verificação da ordem estimada para o modelo. Dentre os métodos para a determinação da ordem destacam-se o do cancelamento entre pólos e zeros e o teste de uma função de perda. Com a análise dos pólos e zeros podemos concluir, no caso de coincidência entre alguns destes, que o modelo obtido foi identificado com a ordem especificada acima da necessária. Em relação à função de perda, diversas identificações são realizadas com modelos de ordem diferentes, calculando-se para cada um destes, uma função baseada nos resíduos do método de identificação aplicado, sendo escolhida a ordem do modelo que a minimize [Isermann, 80].

Em função do resultado da etapa de validação, pode-se constatar a necessidade da aplicação de novos algoritmos de identificação e até mesmo, a realização de novos experimentos de aquisição de dados, como ilustra a Fig. 2.2, que apresenta um esquema do procedimento geral de identificação de sistemas. Com base nesta figura, podemos destacar a característica iterativa do procedimento de identificação, a interdependência de todas as suas etapas e a importância do conhecimento prévio do sistema e da aplicação pretendida ao modelo.

2.3 TRATAMENTO DE SINAIS

O tratamento de sinais é um procedimento que não está diretamente relacionado às etapas de identificação de sistemas. Entretanto, a presença de ruídos nos sinais, é uma característica que pode comprometer o desempenho dos algoritmos de identificação, como ilustra a Fig. 2.3, onde

os ruídos acrescidos ao sinal de saída do sistema degradam a avaliação do erro entre a saída do sistema e a do modelo.

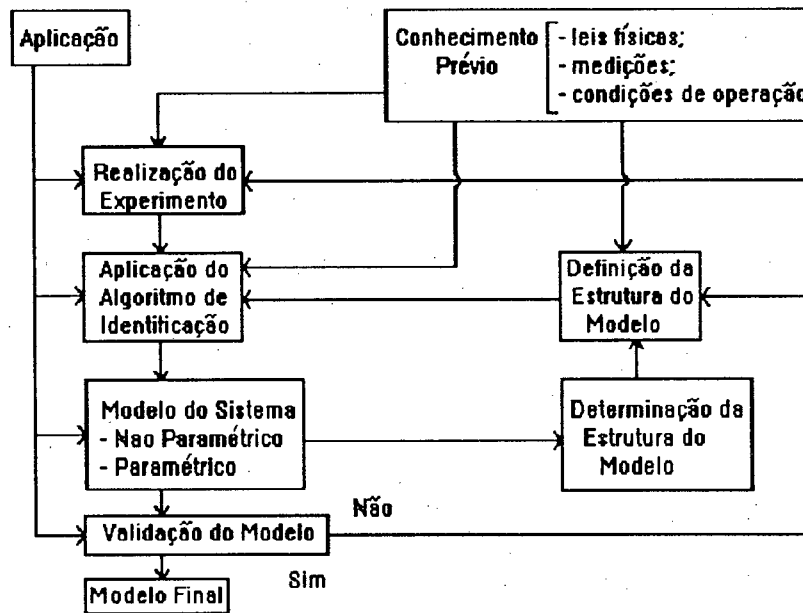


Fig. 2.2 - Procedimento Geral de Identificação de Sistemas [Isermann, 80]

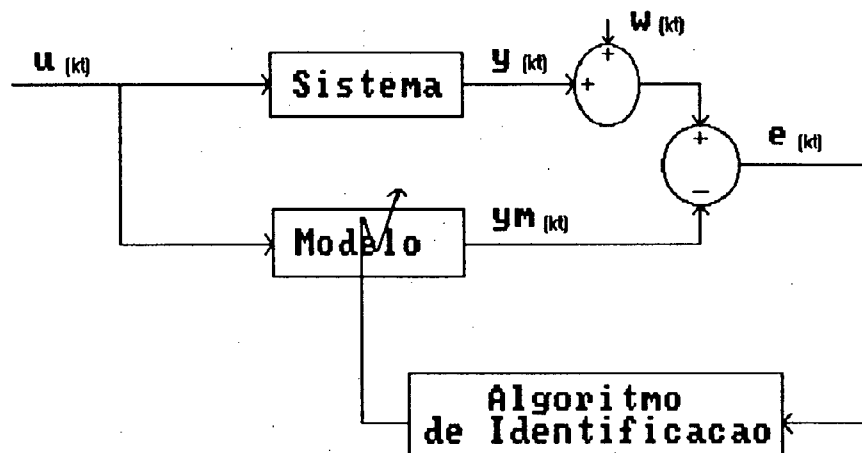


Fig. 2.3 - Influência do ruído no procedimento de Identificação

Desta forma, torna-se necessária a filtragem dos dados de entrada e de saída do sistema, que pode ser realizada durante o procedimento de

aquisição de dados, através de filtros específicos para este fim. Entretanto, de uma maneira geral, quando o procedimento de identificação não for realizado em tempo real, os dados obtidos do experimento podem ser mais criteriosamente tratados, ou seja, surge a possibilidade de aplicação de uma análise prévia, o que facilita a escolha do tipo de tratamento e a especificação dos filtros a serem utilizados. Além disto, todo o procedimento pode ser realizado de maneira interativa, sendo que os resultados obtidos numa etapa podem ser avaliados e utilizados para projetar filtros com características de resposta, a cada passo, mais adequadas ao problema em questão.

2.4 CONCLUSÃO

Com base na descrição das etapas do procedimento de identificação, realizada neste capítulo, podemos concluir que a obtenção de um modelo representativo do comportamento dinâmico de um sistema não é uma tarefa simples e de uma única solução. Na realidade, todo o procedimento é iterativo. Vários algoritmos de identificação podem ser aplicados para a obtenção dos parâmetros do modelo, assim como o tipo de modelo e a sua estrutura podem ser revistos em função dos resultados obtidos.

Um aspecto observado em todas as etapas de um procedimento de identificação e que pode facilitar e diminuir o número de iterações neste processo, é o conhecimento prévio de algumas características do sistema. Este fato permite vislumbrar-se a utilização de uma abordagem híbrida (leis físicas e identificação) na modelagem de sistemas.

Dentro deste contexto, para viabilizar eficazmente a realização de identificação de sistemas, torna-se adequada a utilização de pacotes computacionais, voltados à execução de cada uma das etapas envolvidas neste procedimento. Com este objetivo, foi desenvolvido o programa ISAC - Identificação de Sistemas Assistida por Computador, cujas especificações são apresentada no próximo capítulo.

CAPÍTULO 3

ESPECIFICAÇÃO DO PROGRAMA ISAC

3.1 INTRODUÇÃO

No presente capítulo, apresenta-se a especificação funcional do programa Identificação de Sistemas Assistida por Computador - ISAC. Inicialmente, são levantados os requisitos necessários à realização do procedimento de identificação com auxílio de um pacote computacional, sendo definido o campo de atuação e a forma de processamento do programa. Com base nestes requisitos, é definida a estrutura modular do programa, sendo especificadas as funções inerentes a cada módulo e a forma de comunicação entre eles.

3.2 REQUISITOS DO PROGRAMA

Um dos requisitos básicos do programa é o de possibilitar a identificação de sistemas reais (físicos), sendo consideradas todas as etapas envolvidas neste procedimento.

Inicialmente, como característica fundamental, o programa deve englobar uma base de métodos de identificação, onde os modelos dos sistemas podem ser obtidos utilizando-se diferentes abordagens.

Para a etapa de aquisição de dados torna-se necessária a geração de sinais de entrada, a serem utilizados na excitação do sistema, e a leitura e armazenamento dos dados de saída resultantes.

A etapa de validação também deve ser abrangida pelo programa, devendo ser desenvolvidas ferramentas para a avaliação do desempenho dos modelos identificados na representação do comportamento entrada-saída do sistema.

Embora não seja um requisito essencial ao procedimento de identificação de sistemas, pode-se demonstrar que a maioria dos algoritmos são tendenciosos na presença de ruídos [Isermann, 74; Saridis, 74; Söderström, 74], sendo desejável a realização de um pré-tratamento dos sinais entrada-saída, com vistas a um melhor desempenho dos mesmos.

Além da identificação de sistemas reais, o programa deve poder também ser utilizado no ensino de técnicas de identificação, considerando-se a simulação de todas as experiências envolvidas neste procedimento, incluindo a geração de sinais, simulação de sistemas, identificação sem e com ruído, validação, etc.

Por fim, outro requisito essencial do programa é o desenvolvimento de uma interface de alto nível com o usuário, englobando o traçado de gráficos, o fornecimento de informações sobre parâmetros de entrada, a apresentação de resultados e uma estrutura de menus, para o controle das suas diversas funções.

3.3 ESPECIFICAÇÃO DO PROGRAMA

Inicialmente, considerando a variedade de tipos de sistemas existentes, e o número de algoritmos de identificação, restringe-se a abrangência do programa a uma classe específica de sistemas, sendo prevista na especificação da sua estrutura a possibilidade de expansão a outros casos. Assim, o ISAC será voltado à identificação de sistemas monovariáveis, invariantes no tempo, a parâmetros concentrados, lineares ou linearizáveis em torno de um ponto de funcionamento. Isto permite que uma grande parte dos sistemas físicos reais estudados pela teoria de controle seja tratada pelo programa. Desta forma, em relação à etapa de definição da classe de modelos, caberá ao usuário verificar se o sistema em questão enquadra-se no escopo do programa.

Definido o escopo de abrangência, o primeiro passo para a especificação da estrutura do programa é a escolha da forma de

processamento do procedimento de identificação: em tempo real ou diferido. O processamento em tempo real, somente é indispensável quando o modelo identificado é utilizado em algoritmos de controle "on-line", ou quando não se dispõe de suporte para o armazenamento dos dados obtidos no experimento. Sendo assim, optou-se pelo processamento em tempo diferido, permitindo-se a análise e pré-tratamento dos sinais e, a aplicação de diferentes algoritmos sobre a mesma série de dados.

Analizando o procedimento de identificação de sistemas, descrito no Capítulo 2, destacamos as suas etapas, que apesar de interdependentes, podem ser executas de maneira isolada por uma ou mais rotinas específicas, como por exemplo, rotinas para a geração de sinais, a aplicação dos algoritmos de identificação, e rotinas para a validação dos modelos identificados.

Desta forma, na definição da estrutura do ISAC, todos os seus módulos são especificados de forma a executarem suas funções independentemente, interagindo exclusivamente com o usuário e com uma estrutura de "arquivos", responsável pelo armazenamento dos dados do sistema.

Além das facilidade de implementação, a utilização desta abordagem, assegura a modularidade do programa, possibilitando a inserção de novos algoritmos, sem maiores modificações na sua estrutura.

A estrutura modular especificada para o programa é apresentada na Fig. 3.1. Basicamente, este é composto por um módulo gerenciador, pelos módulos de entradas, saídas e gerenciamento de arquivos, os quais constituem o ambiente de execução do programa, e pelo módulo de ferramentas, onde são executadas as funções inerentes ao procedimento de identificação.

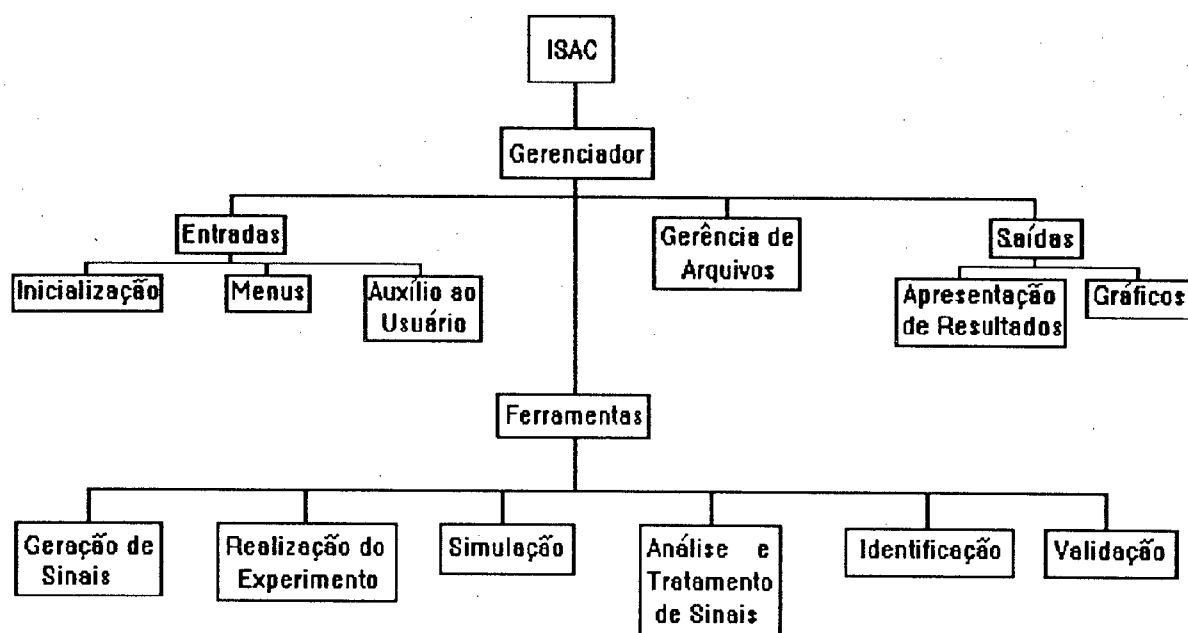


Fig. 3.1 - Estrutura dos módulos do programa

3.3.1 MÓDULO DE GERENCIAMENTO

O módulo de gerenciamento será o responsável pelo controle geral do programa e seleção das tarefas a serem realizadas, coordenando as atividades de todos os demais módulos.

Em função da seleção pelo usuário das rotinas a serem executadas, o *gerenciador* deverá ativar o respectivo módulo, coordenando a sua execução no que se refere a alocação de memória e tratamento de erros, e restabelecendo a configuração do programa ao término da mesma.

Em relação ao tratamento de erros, quando a área de dados do programa estiver saturada, deverão ser desabilitadas as rotinas que se utilizem de novas áreas de memória, comunicando-se ao usuário para que este venha a liberar espaço, através do descarte de dados já utilizados. Da mesma forma, o gerenciador deve verificar se os parâmetros fornecidos

pelo usuário são coerentes em relação aos dados requeridos pela rotina em execução.

3.3.2 MÓDULO DE ENTRADAS

Observando o princípio da modularidade, e da independência entre os módulos do sistema, a entrada de parâmetros para a execução de um algoritmo específico, por exemplo, a geração de um sinal, deverá ser realizada pelo módulo relacionado ao algoritmo em questão.

Desta forma, resta ao módulo de entradas a criação e edição de arquivos com os dados gerais do sistema, abrangendo modelos e sinais de entrada e saída. A especificação destas funções, principalmente em relação à edição de arquivos de sinais, prevê a utilização do programa em casos onde os dados de entrada e saída do sistema são obtidos manualmente.

Além da entrada de dados, são consideradas no módulo de entradas as funções de interface com o usuário, relacionadas à seleção das rotinas a serem executadas e ao auxílio ao usuário. Para tanto, deve ser desenvolvida uma estrutura de menus, onde o usuário selecione a função a ser executada, e uma "função ajuda", onde sejam apresentadas informações sobre a utilização do programa direcionadas ao nível de execução em que este se encontra.

3.3.3 MÓDULO DE SAÍDAS

O módulo de saídas deve realizar as funções relacionadas à apresentação dos resultados obtidos e armazenados no programa, englobando o traçado dos gráficos e a visualização dos modelos obtidos. Na especificação desta versão do ISAC, não considera-se a utilização, direta, de uma impressora para a apresentação dos resultados, sendo prevista a construção de arquivos do tipo texto para a impressão dos resultados com comandos do sistema operacional.

3.3.4 MÓDULO DE GERÊNCIA DE ARQUIVOS

Para viabilizar a execução de todos os procedimentos envolvidos na identificação de sistemas, de forma independente, especifica-se uma estrutura de "arquivos", onde serão armazenados os principais tipos de dados do programa, ou seja, os sinais de entrada e saída e os modelos dos sistemas.

A estrutura de arquivos é considerada no programa segundo duas abordagens. Inicialmente, em função do tempo despendido nas operações de interface com dispositivos de memória de massa, todos os sinais e modelos gerados pelo programa ficarão armazenados em sua área de dados, podendo ser imediatamente acessados por todos os seus módulos. A estas estruturas de armazenamento de dados na memória primária, resolveu-se denominar "arquivos virtuais". A estrutura de arquivos considera ainda os arquivos em memória secundária, possibilitando a recuperação de sinais e modelos anteriormente criados pelo programa, quando da sua reinicialização. Além disto, os sinais obtidos por outros sistemas de geração e aquisição de dados, desde que devidamente formatados aos padrões definidos pelo programa, poderão ser neste inseridos para a realização do procedimento de identificação.

Todas as funções inerentes à estrutura de arquivos do programa serão realizadas pelo módulo de gerência de arquivos, que será responsável pelo gerenciamento da área de dados do programa e pela interface desta com dispositivos de memória de massa. Desta forma, são especificadas funções para a leitura de um arquivo em memória permanente, para o salvamento de arquivos, e de verificação dos arquivos residentes no programa e no diretório corrente. Além disto, devem ser implementadas funções de seleção de um arquivo para a utilização nos diversos algoritmos dos demais módulos.

3.3.5 MÓDULO DE FERRAMENTAS

O módulo de ferramentas será responsável pela execução do procedimento de identificação, sendo formado por um conjunto de módulos relacionados diretamente às seguintes funções: geração de sinais e simulação de sistemas; realização do procedimento de aquisição de dados; análise e tratamento de sinais, aplicação dos algoritmos de identificação e validação dos modelos identificados.

3.3.5.1 Módulo de Geração de Sinais

O programa deverá possibilitar a geração de diversos sinais de excitação a serem utilizados na realização do experimento de aquisição de dados e na sua simulação.

Na descrição das etapas do procedimento de identificação ressaltamos a importância da aplicação de um "sinal de excitação persistente" para a obtenção de um modelo representativo do sistema. Um dos sinais mais eficientes na excitação de todos os modos de um sistema é a sequência binária pseudo-aleatória (SBPA), que apresenta um espectro em frequência bastante largo e com uma função autocorrelação que se aproxima bem da função delta de Dirac [Cardoso F., 79].

O degrau é um sinal que embora não apresente boas características em frequência, é muito utilizado na análise do comportamento de sistemas. Além disto, encontram-se alguns algoritmos clássicos de identificação baseados na resposta do sistema a uma excitação do tipo degrau.

Na especificação do módulo de identificação, a ser discutida nas próximas seções, considera-se a utilização do método da correlação, resultando numa primeira etapa num modelo não paramétrico; a sequência ponderada; que é a versão discreta da resposta ao impulso do sistema. Com a excitação do sistema com um sinal do tipo pulso unitário, a sequência ponderada pode ser obtida diretamente. De qualquer forma,

quando este procedimento não puder ser utilizado, por razões relacionadas à operação do sistema, a validação do modelo com um pulso unitário serve como medida de desempenho do algoritmo em questão.

Desta forma, no módulo de geração de sinais, considerar-se-a a geração da sequência binária pseudo-aleatória, do degrau, do pulso, e de sinais senoidais, estes últimos, por serem comumente utilizados na excitação de sistemas.

3.3.5.2 Módulo de Realização do Experimento

O módulo de realização do experimento deverá possibilitar, através de uma placa de aquisição de dados, a inserção de um sinal de excitação em um sistema e a leitura e armazenamento dos dados de saída resultantes.

Além disto, o programa deverá permitir que arquivos de sinais de entrada e saída, obtidos por outros programas de aquisição de dados sejam inseridos no contexto do programa, para a identificação.

3.3.5.3 Módulo de Simulação

Quando utilizado em ambiente acadêmico, o programa deverá permitir a simulação da etapa de aquisição de dados, considerando diversos aspectos práticos envolvidos neste procedimento.

O módulo de simulação de sistemas será o responsável pela simulação do comportamento de um sistema em função de um sinal de excitação específico, sendo também consideradas ferramentas para a geração e adição de ruídos brancos ou coloridos ao sinal de saída resultante.

3.3.5.4 Módulo de Análise e Tratamento de Sinais

Para o tratamento de sinais de entrada e saída do sistema, duas abordagens deverão ser consideradas. Quando a presença do ruído sobre o

signal é facilmente detectável prevê-se o tratamento iterativo de uma curva, através de técnicas de regressão e substituição de pontos espúrios. Uma abordagem mais complexa será a utilização de ferramentas para a análise do espectro em frequência dos sinais do sistema, e o projeto de filtros digitais a serem aplicados sobre estes.

3.3.5.5 Módulo de Identificação

O módulo de identificação, englobará os algoritmos para a identificação do modelo do sistema, sendo previstos, inicialmente, algoritmos baseados em três diferentes abordagens: no princípio dos mínimos quadrados, na correlação entre os sinais do sistema e nas característica de resposta ao degrau.

Os algoritmos de predição de erro, baseados no princípio dos mínimos quadrados, são algoritmos paramétricos, onde a estrutura do modelo a ser identificado, deve ser conhecida ou arbitrada, para posterior verificação. Dentro desta classe de algoritmos considera-se a implementação dos seguintes métodos: mínimos quadrados simples, mínimos quadrados generalizados; variáveis instrumentais e método da matriz estendida.

O método da correlação, além de apresentar um bom desempenho na presença de ruídos, permite a visualização de um modelo não paramétrico, a sequência ponderada do sistema. A sequência ponderada pode servir de base para a definição da estrutura do modelo [Oppenheim, 83], na aplicação dos métodos paramétricos, quando as informações prévias sobre o sistema não forem suficientes para tanto.

Numa abordagem acadêmica, e prevendo-se a sua utilização em muitos casos práticos de pouca complexidade é considerada a implementação dos algoritmos de identificação baseados na resposta do sistema ao degrau, os quais também podem servir de base para os demais métodos, na definição de ganhos, constantes de tempo dominantes, etc. Ao contrário dos algoritmos de estimação de parâmetros, desenvolvidos para

um caso geral, os métodos gráficos baseiam-se numa estrutura pré-definida para o modelo, e seu desempenho na representação do sistema, está diretamente relacionado à adequação desta estrutura ao sistema a ser identificado.

3.3.5.6 Módulo de Validação

Neste módulo, serão consideradas as ferramentas para a validação do modelo identificado, envolvendo o desempenho deste na representação do comportamento entrada-saída do sistema, e a avaliação da estrutura especificada para o mesmo quando da aplicação dos algoritmos de identificação.

O modelo deverá poder ser simulado com a entrada do sistema, sendo o resultado comparado com a sua saída real, num procedimento que deve considerar critérios de erros e a análise gráfica dos resultados.

Em relação à estrutura do modelo, o módulo de validação deve abranger rotinas que permitam a pré-determinação da ordem e do atraso do sistema, e a comprovação final dos mesmos após a identificação.

3.4 CONCLUSÃO

Em função da independência, em relação à execução, das etapas do procedimento de identificação de sistemas, todo o programa foi especificado como um conjunto de rotinas desenvolvidas em separado e reunidas num ambiente de alto nível.

O ambiente de execução do programa, será o responsável pelo interfaceamento dos seus diversos módulos, devendo proporcionar ao usuário um conjunto de ferramentas auxiliares, possibilitando a entrada de dados, a apresentação de resultados, a geração de gráficos e o gerenciamento da área de dados do programa, a qual será responsável pela interface das rotinas acima mencionadas.

Em relação às especificações, devemos ressaltar ainda, que embora o escopo de abrangência do programa seja limitado, toda a estrutura especificada prevê, desde a sua concepção, a expansão de seus módulos possibilitando a inserção de novas ferramentas e algoritmos.

CAPÍTULO 4

IMPLEMENTAÇÃO DO ISAC

4.1 INTRODUÇÃO

Descritos os requisitos e especificada a estrutura dos módulos do programa, passamos neste capítulo, ao detalhamento da sua implementação.

Desta forma, são apresentadas as principais funções do ambiente de execução do programa e as rotinas implementadas para a geração de sinais, simulação de sistemas e análise e tratamento de sinais. Por fim, são descritos os algoritmos de identificação implementados no programa e as rotinas para a validação dos modelos identificados.

A descrição dos algoritmos implementados se restringirá aos seus fundamentos e às suas principais características, uma vez que o objetivo deste trabalho é a implementação de um pacote que auxilie às principais etapas da identificação de sistemas e não o estudo teórico de seus métodos.

4.2 O AMBIENTE DE EXECUÇÃO DO PROGRAMA

É constituído por um conjunto de funções que objetivam dar suporte, dentro de uma interface amigável, à execução do programa, através do gerenciamento de sua estrutura de dados e de rotinas de interface com o usuário, de entrada de dados e apresentação de resultados

4.2.1 ESTRUTURA DE DADOS DO PROGRAMA

O programa trabalha basicamente com arquivos de sinais de entrada e saída e arquivos de modelos, que ficam residentes em sua área de dados

(arquivos virtuais) e podem ser armazenados em dispositivos de memória permanente.

4.2.1.1 Diretório de Arquivos Virtuais

Objetivando minimizar o tempo de acesso aos arquivos durante a execução do programa, é estabelecido que todas as suas rotinas operem diretamente sobre um "diretório de arquivos virtuais" que em termos práticos, funciona como um diretório do sistema operacional.

O programa trabalha com modelos contínuos e discretos, representados pelas equações (4.1) e (4.2) respectivamente,

$$G(S) = \frac{(b_0 + b_1S + b_2S^2 + \dots + b_nS^n) e^{-aS}}{a_0 + a_1S + a_2S^2 + \dots + a_nS^n} \quad (4.1)$$

$$G(z^{-1}) = \frac{(b_0 + b_1z^{-1} + b_2z^{-2} + \dots + b_nz^{-n}) z^{-a}}{1 + a_1z^{-1} + a_2z^{-2} + \dots + a_nz^{-n}} \quad (4.2)$$

onde a é o atraso de transporte do sistema. Em ambos os tipos de modelos, para o seu armazenamento é utilizada uma estrutura de arquivo virtual, que na realidade é um descritor de arquivo, sendo registrados os coeficientes das equações, o atraso, e para o modelo discreto, o período de amostragem com o qual este foi obtido.

Para o armazenamento dos sinais de um sistema, considerando as variações de tamanho que os arquivos virtuais apresentariam em função do número de pontos, foram implementadas duas estruturas auxiliares, uma para os sinais de entrada e outra para os de saída. Nestas estruturas, os dados de todos os sinais criados ou inseridos no programa, são gravados sequencialmente. Nos arquivos virtuais, propriamente ditos, são armazenadas as informações sobre o número de pontos que compõe o sinal, o período de amostragem utilizado na sua obtenção e indicadores de endereço, os quais apontam para a posição de seus dados dentro das referidas estruturas.

A Fig. 4.1 ilustra a implementação do diretório de arquivos virtuais do programa, onde podemos observar dois arquivos virtuais de sinais de entrada, um de saída, e a correspondente posição de seus dados dentro das estruturas de armazenamento. Da mesma forma encontramos um arquivo virtual de modelo, que não se utiliza de nenhuma estrutura auxiliar. A implementação da estrutura do diretório de arquivos virtuais é transparente ao usuário, sendo-lhe apresentado apenas os nomes dos arquivos.

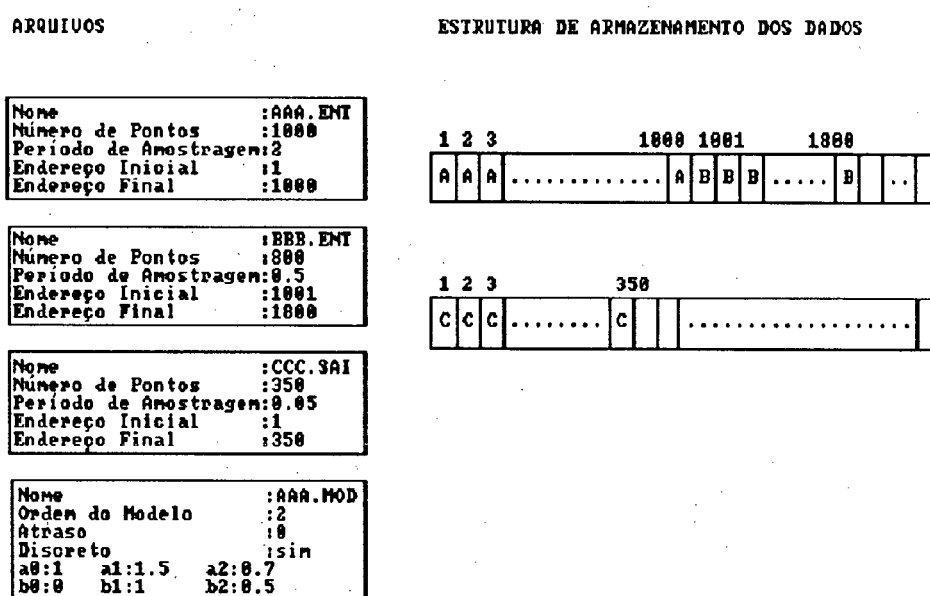


Fig. 4.1 - Estrutura do Diretório de Arquivos Virtuais

4.2.1.2 Arquivos em Memória de Massa

Nos arquivos de memória de massa, tanto os modelos como os sinais de entrada e saída, são gravados em estruturas do tipo "file of real", sendo armazenadas seqüencialmente as informações sobre as características do arquivos e os dados do mesmo.

Como forma de controle, estabelecemos que os arquivos utilizados no programa devem ter as extensões .ENT, .SAI, e .MOD, para arquivos de

sinais de entrada, saída e modelos respectivamente. No caso de um arquivo gerado pelo programa, a extensão do nome é acrescentada automaticamente.

4.2.1.3 Gerenciamento da Estrutura de Dados

Em relação à estrutura de dados do programa, são implementadas funções que coordenam o diretório de arquivos virtuais, através do controle sobre o número máximo de arquivos que podem ser criados (20 arquivos de sinais e 15 de modelos), e do gerenciamento da alocação dos arquivos de sinais dentro da estrutura de armazenamento de dados. Além disto, sobre o diretório de arquivos virtuais, foram desenvolvidas funções de seleção, relacionadas com a extensão do nome do arquivo, sendo estas funções acessíveis a todas as rotinas do programa que se utilizam de sinais e modelos. Desta forma, quando, por exemplo, uma rotina solicita o nome de um arquivo de sinal de entrada, o usuário pode digitar o nome do arquivo ou solicitar a seleção, que é realizada em uma janela onde todos os arquivos de sinais de entrada residentes no diretório do programa são apresentados.

Em relação à interface entre o diretório de arquivos virtuais e dispositivos de memória permanente, foram desenvolvidas funções que permitem a visualização dos dois diretórios, carregar para a memória, salvar em disco ou simplesmente descartar os arquivos não mais utilizados.

4.2.2 A INTERFACE COM O USUÁRIO

No desenvolvimento da interface com o usuário, foi implementada uma estrutura de menus que permite a seleção das diversas tarefas a serem executadas nos diferentes módulos do programa. Além disto, durante a execução dos algoritmos, são apresentadas informações ou gabaritos que objetivam orientar o usuário na entrada dos parâmetros requeridos por estes. Para completar o auxílio ao usuário foi desenvolvida uma rotina de "ajuda" que apresenta informações sobre o algoritmo a ser utilizado. O

conteúdo das informações possibilita ao usuário um complemento às interfaces do programa, não apenas em relação à forma de entrada dos dados, mas também na orientação sobre um melhor aproveitamento dos algoritmos. As telas geradas pelo programa ilustrando as características acima descritas são apresentadas na Fig. 4.2. Os detalhes de implementação da estrutura de menus e da rotina de auxílio ao usuário são apresentados no manual do usuário em anexo.

4.2.3 ENTRADA DE DADOS E APRESENTAÇÃO DE RESULTADOS

Nos casos em que o experimento de identificação tenha sido realizado com a leitura manual dos dados do sistema, o ambiente do programa permite a criação de arquivos de sinais e edição direta de seus dados através do teclado. Da mesma forma, foram desenvolvidas funções para a criação e edição de arquivos de modelos discretos e contínuos, as quais foram acrescentadas ao módulo de simulação, uma vez que são utilizadas para a definição dos modelos a serem simulados.

Na apresentação dos resultados o ambiente permite a geração de gráficos dos arquivos de sinais e a visualização dos seus dados e dos arquivos de modelos em telas dentro do programa. No traçado de gráficos foram implementadas funções para a ampliação de uma faixa do sinal (função zoom) e a verificação dos valores dos pontos traçados (função cursor). A visualização de um arquivo do programa, pode ser realizada tanto na tela como em arquivos do tipo texto, sendo apresentados todos os dados de um sinal e os parâmetros de um modelo. Os arquivos gerados por esta rotina podem ser diretamente tratados por comandos do sistema operacional. A Fig. 4.3 apresenta o gráfico de um arquivo de sinal e a correspondente listagem de seus dados numa tela do programa.

ISAC

PARAMETROS PARA IDENTIFICACAO		OPCOES
Arquivo com o Modelo do Processo	MODELO .MOD	Gerar sinais
Arquivo com o sinal de Entrada	SB511 .ENT	Simular processos
Arquivo com o sinal de Saída	SIS2SB20.SAI	IDENTIFICAR PROCESSOS
Numero de Pontos (max = 1000)	100	Editar arquivos
Ordem do modelo (max=10)	2	Tracar graficos
Atraso do modelo (em Periodos)	0	Analisar e tratar sinais
O processo tem Transferencia Direta? (S/N)		apresentar Resultados
	ACE	Fim
		METODOS
		MINIMOS QUADRADOS SIMPLES
		Minimos quadrados generalizados
		variaveis Instrumentais
		matriz Estendida
		Correlacao
		metodos Graficos
		OPCOES
		polos e Zeros de um modelo
		Validar modelos
		Pre-determinacao da ordem

11 Ajuda 12 Diretorio 13 Carrega 14 Salva 15 Apaga

ISAC

AJUDA		OPCOES
MINIMOS QUADRADOS SIMPLES - Obtem o modelo de um processo atraves do algoritmo recursivo minimos quadrados. Para a identificacao de um processo algumas informacoes sobre este devem ser conhecidas ou arbitradas. Estas informacoes sao a ordem do modelo, o atraso (periodos) e se o processo apresenta transferencia direta ou nao, o que implica num modelo com ou sem o termo b0. Estas informacoes podem ser inseridas via teclado ou com a selecao de um arquivo resultante de outro metodo que venha a ser utilizado como parametro inicial.		Gerar sinais Simular processos IDENTIFICAR PROCESSOS arquivos raficos e tratar sinais ar Resultados
ACE ETRON 11-1000		ODOS DE SINALIS os generalizados umentais a s OES e um modelo
		Pre-determinacao da ordem

11 Ajuda 12 Diretorio 13 Carrega 14 Salva 15 Apaga

Fig. 4.2 - Telas de Interface do programa ISAC

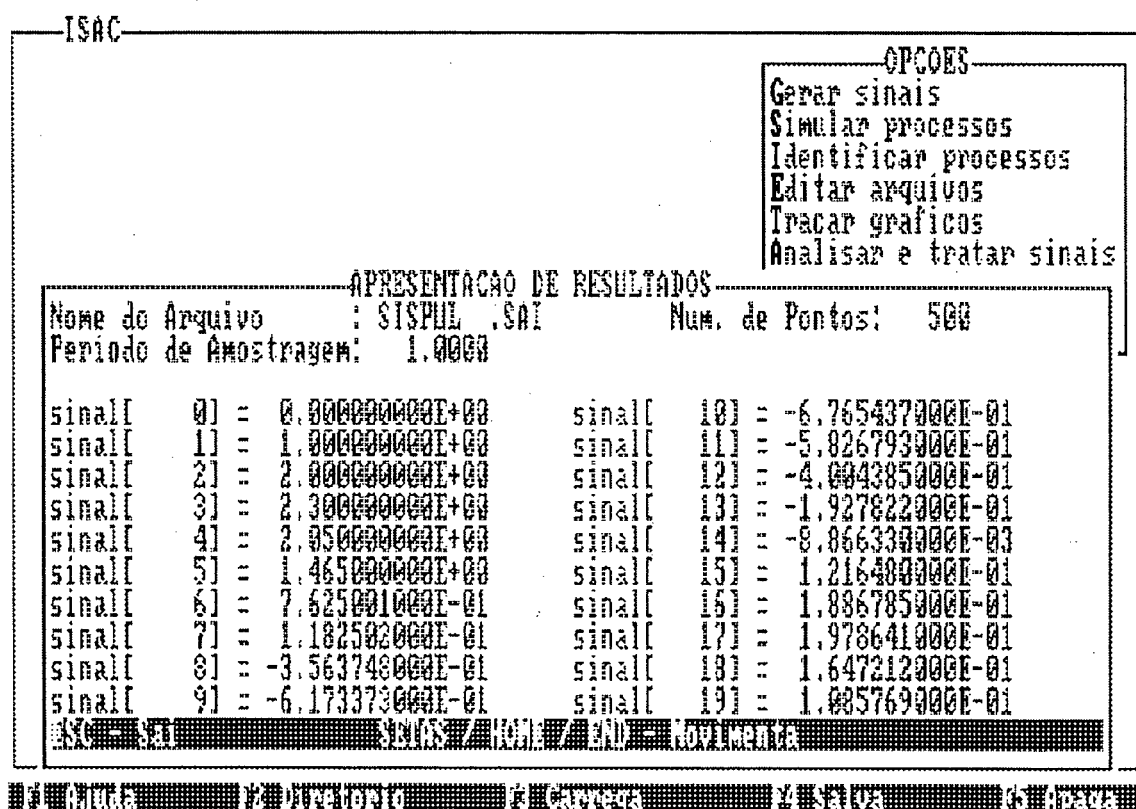
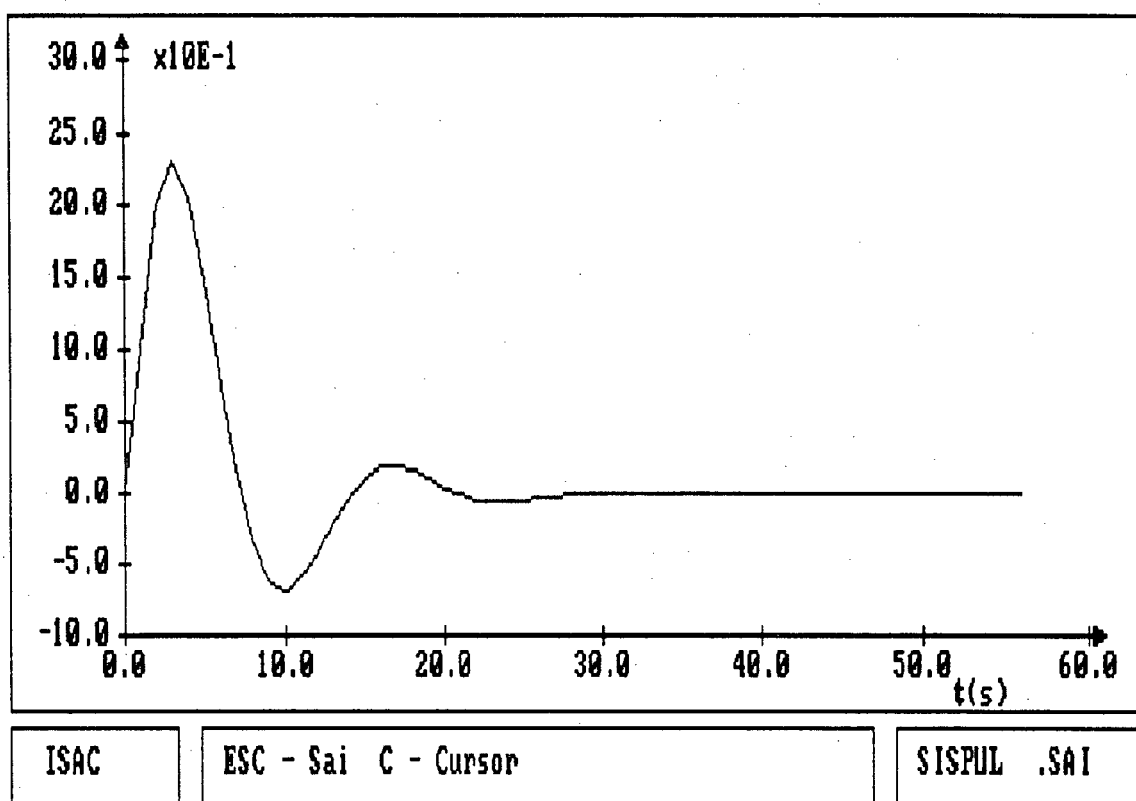


Fig. 4.3 - Gráfico e listagem de um arquivo de sinal

4.3 GERAÇÃO DE SINAIS

O programa possibilita a geração de diversos sinais de excitação a serem utilizados na realização do experimento e na simulação de sistemas. Desta forma, o módulo de geração de sinais considera, além do degrau e dos sinais sinusoidais, comumente utilizados na excitação de sistemas, sinais binários como a seqüência binária pseudo-aleatória e o pulso, os quais apresentam características especiais dentro do contexto do procedimento de identificação.

Para a geração dos sinais deste módulo, com exceção da seqüência binária pseudo-aleatória que é apresentada a seguir, foram implementadas rotinas de pouca complexidade, destacando-se apenas que o programa permite a geração de sinais compostos pelo somatório de até oito senóides. A Fig. 4.4 apresenta o gráfico de um sinal com quatro componentes de freqüências distintas, gerado pelo programa.

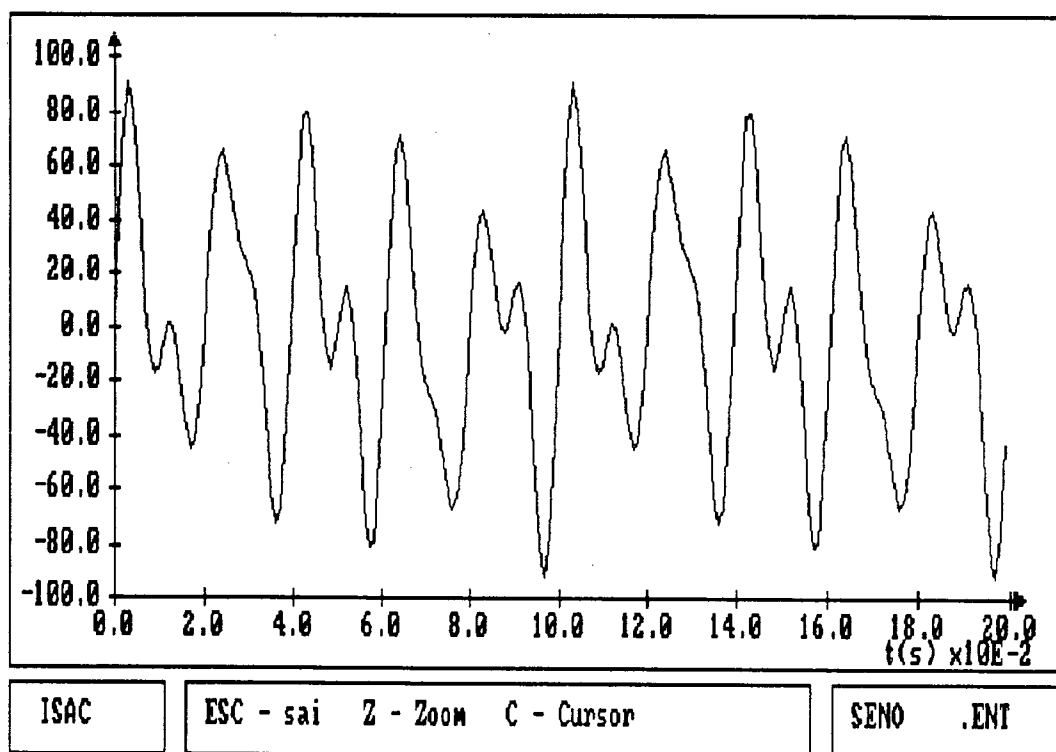


Fig. 4.4 - Sinal gerado pelo programa

4.3.1 SEQÜÊNCIA BINÁRIA PSEUDO-ALEATÓRIA

A implementação da SBPA é feita com circuitos digitais, utilizando apenas registradores de deslocamento e funções lógicas, ou em computadores digitais, através da simulação destes circuitos. Em função do número de registradores ou variáveis utilizadas na sua geração (n), podemos obter seqüências de diferentes comprimentos (l), que são calculados pela seguinte relação:

$$l = 2^n - 1 \quad (4.3)$$

Como exemplo, a Fig. 4.5 apresenta o esquema básico de um circuito para a geração de uma SBPA utilizando-se quatro registradores de deslocamento.

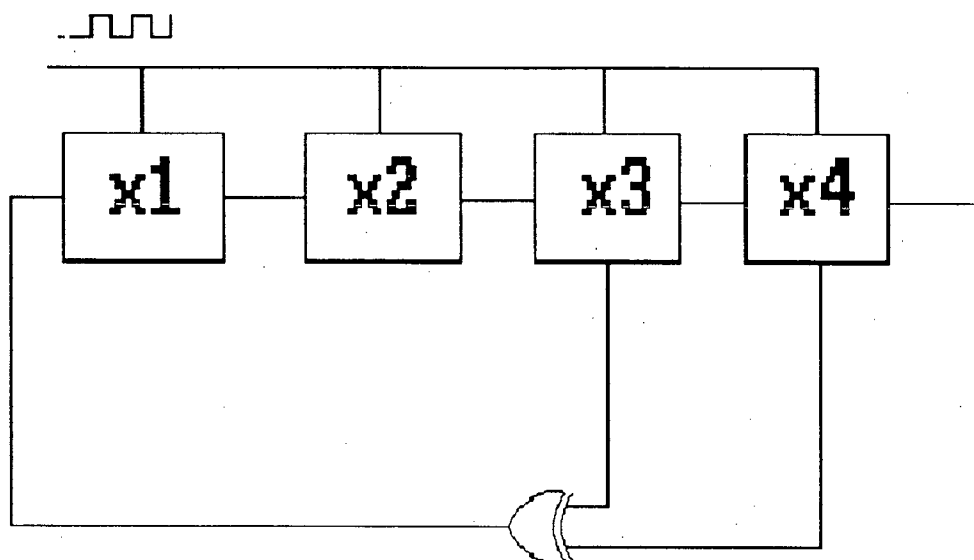


Fig. 4.5 - Circuito para a geração de um SBPA

[Cardoso F., 79]

Considerando que o estado inicial dos registradores é 1111, obtém-se a seqüência de comprimento máximo:

1 1 1 1 0 0 0 1 0 0 1 1 0 1 0 1 1 1 1 ...

$$l = 2^4 - 1 = 15$$

Uma variação da SBPA é a seqüência binária negativa, que ao invés de gerar os níveis 1 e 0, gera elementos em 1 e -1, resultando num valor médio nulo quando o número de pontos tende ao infinito [Cardoso F., 79].

A escolha do comprimento máximo da seqüência está relacionada à maior constante de tempo que se deseja identificar. O seu produto pelo período de amostragem deve ser no mínimo 10 vezes maior do que o tempo necessário para a resposta do sistema atingir a $t_{95\%}$, ou seja, aproximadamente o regime permanente [Martinez, 70]. Por exemplo, se $t_{95\%}$ do sistema é 100 segundos e o período de amostragem do sinal é 1 segundo, torna-se necessária a utilização de um seqüência de comprimento mínimo igual a 1023 ($2^{10}-1$).

O programa permite a geração de seqüências binárias (negativas) de até 10000 pontos, podendo ser selecionados comprimentos entre 15 e 2047 pontos. A amplitude da seqüência é especificada pelo usuário segundo as características do sistema a ser identificado, de forma a não afetar o seu funcionamento normal. Como exemplo, a Fig. 4.6 apresenta o gráfico de uma SBPA de amplitude 10 e comprimento máximo igual a 15, podendo-se verificar a sua periodicidade a partir do 16º ponto.

4.4 REALIZAÇÃO DO EXPERIMENTO DE AQUISIÇÃO DE DADOS

Para a realização do experimento de aquisição de dados, o programa considera em sua estrutura, um módulo responsável pela excitação do sistema e aquisição do sinal de saída resultante.

Este módulo, deve funcionar, em princípio, como uma interface entre o diretório de arquivos virtuais do programa e a placa de aquisição de dados. O sinal de excitação, deve ser previamente gerado no módulo de geração de sinais, ficando disponível no diretório de arquivos, onde também serão gravados os dados do sinal de saída a ser obtido. Através de interrupções, a cada período de amostragem, definido pelo usuário, respeitando-se as limitações do hardware da placa de interface com o

sistema, aplica-se a este, o sinal de excitação, medindo-se o correspondente sinal de saída. A Fig. 4.7, apresenta a estrutura do módulo de realização do experimento de aquisição de dados.

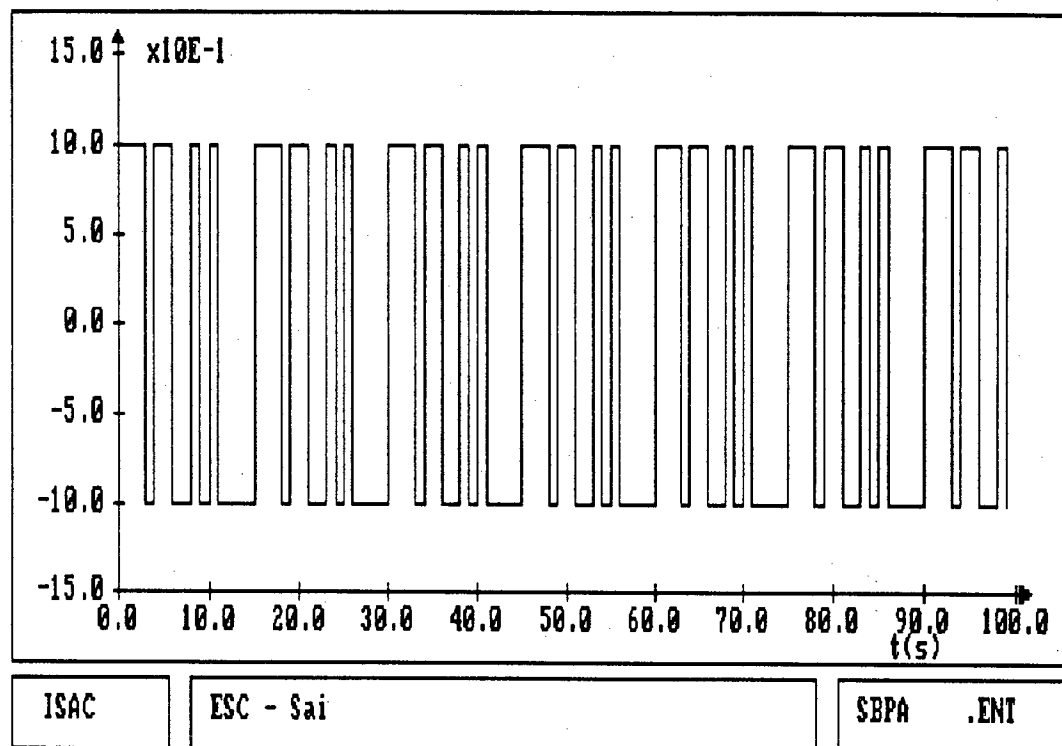


Fig. 4.6 - Gráfico de uma SBPA gerada pelo programa

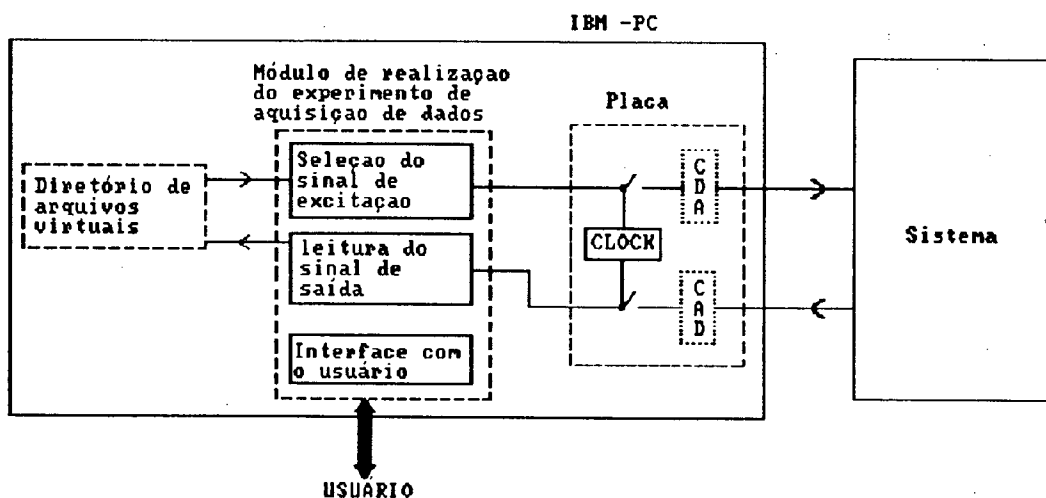


Fig. 4.7 Estrutura do módulo de realização do experimento de aquisição de dados

A implementação das rotinas, depende diretamente do tipo de placa de aquisição de dados a ser utilizada. Em virtude da não disponibilidade desta, o desenvolvimento do módulo de realização do experimento não foi considerado nesta primeira versão do programa ISAC, ficando como proposta para futuros trabalhos. Entretanto, a utilização do programa para a identificação de sistemas reais não deve ficar prejudicada, uma vez que, o experimento de aquisição de dados pode ser executado, de forma independente, por outros pacotes computacionais e suportes de hardware.

4.5 SIMULAÇÃO DE SISTEMAS

Na especificação do programa, além da identificação de sistemas reais, previmos a sua utilização no ensino das técnicas de identificação, sendo considerada na sua estrutura uma rotina para a simulação de sistemas. Além disto, foram desenvolvidas rotinas auxiliares para a criação e discretização de modelos.

4.5.1 SIMULAÇÃO DE MODELOS DISCRETOS

Para a simulação de sistemas optou-se pela utilização de modelos discretos, representados pela equação (4.2), da qual pode-se obter a seguinte equação de recorrência:

$$y(k) = b_0 u(k-a) + b_1 u(k-1-a) + \dots + b_n u(k-n-a) - a_1 y(k-1) - \dots - a_n y(k-n), \quad (4.4)$$

onde: $y(k)$ - saída do sistema no instante k ;
 $u(k)$ - entrada do sistema no instante k ;
 n - ordem do sistema;
 a - atraso de transporte do sistema;
 b_i, a_i - parâmetros do modelo.

A implementação da rotina de simulação se restringe às operações descritas na equação (4.4), uma vez que o período de amostragem do sinal resultante está definido na estrutura do modelo utilizado, e os arquivos contendo o respectivo modelo e o sinal de excitação devem ser gerados

antecipadamente. Além da simplicidade, a opção pela utilização desta abordagem considera ainda, a sua utilização no módulo de validação de modelos, uma vez que a maioria dos algoritmos de identificação utilizados no programa conduzem a modelos discretos.

A rotina de simulação permite ainda a geração e adição de ruídos sobre o sinal de saída do sistema, os quais são obtidos através de um sinal aleatório (ruído branco) que varia entre -1 e 1 [Rabiner, 78]. No caso de não se desejar um ruído "branco", o programa permite a introdução de um filtro, que é aplicado sobre esta sequência para a obtenção de um ruído "colorido". Os parâmetros do filtro, e o percentual de ruído a ser adicionado sobre o valor médio do sinal são definidos pelo usuário, possibilitando-se a simulação de diversas condições de operação para o sistema.

A Fig. 4.8 apresenta um esquema ilustrando o procedimento de simulação de sistemas.

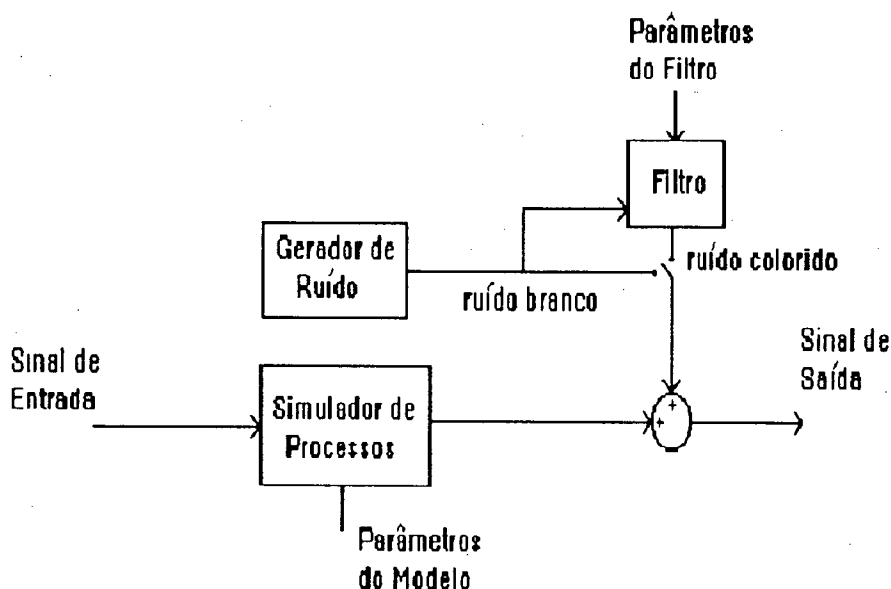


Fig. 4.8 - Esquema do procedimento de simulação

4.5.2 EDIÇÃO DE ARQUIVOS E DISCRETIZAÇÃO DE MODELOS

Foram acrescentadas ao módulo de simulação, rotinas para criação de arquivos de modelos discretos e contínuos e discretização destes últimos.

A criação de arquivos com modelos discretos e contínuos se restringe à edição dos parâmetros das equações (4.1) e (4.2) respectivamente.

Para facilitar a entrada de dados, a rotina para criação de modelos contínuos permite a edição de até 5 (cinco) polinômios para o numerador e denominador da função de transferência do sistema, sendo que o próprio algoritmo encarrega-se da multiplicação dos polinômios para a obtenção do modelo na sua forma geral.

Na discretização de modelos, inicialmente é efetuada a sua transformação para equações de estado na forma canônica controlável. Uma vez tendo-se as equações de estado, o modelo é discretizado por expansão de séries, sendo a função de transferência discreta obtida através do algoritmo de Fadeev-Leverrier [Chen, 70].

4.6 ANÁLISE E TRATAMENTO DE SINAIS

Para a análise e tratamento dos sinais, duas abordagens foram consideradas. A utilização de técnicas de regressão para o tratamento interativo de uma curva e o projeto de filtros digitais para a aplicação sobre os sinais do sistema. À primeira nos referimos como análise e tratamento no tempo e à segunda, por considerar o espectro em frequência dos sinais, como análise e tratamento na frequência.

4.6.1 ANÁLISE E TRATAMENTO DE SINAIS NO TEMPO

Quando o sinal de saída do sistema apresentar uma forma conhecida como, por exemplo, a resposta do sistema ao degrau ou ao pulso, pode-se facilmente verificar a presença de ruídos sobre o sinal, os quais, em função

das condições do experimento de aquisição de dados podem estar associados a toda a curva, ou a parte dela. Além disto, podemos identificar a presença de pontos espúrios oriundos de erros grosseiros. Nestes casos, uma alternativa para o tratamento de sinais é a modificação da curva através da substituição direta de seus pontos. Desta forma, foram desenvolvidas no módulo de análise e tratamento de sinais, rotinas que operam diretamente sobre o gráfico de resposta do sistema e permitem a substituição de pontos isolados e de faixas do sinal pela melhor reta, parábola ou cúbica que passa pelos pontos selecionados. Cumpre-se ressaltar, que a curva original não é alterada, sendo que a rotina cria um novo arquivo, onde são armazenados os dados do sinal tratado.

Em relação aos pontos espúrios, após a sua identificação pelo usuário com um cursor sobre a curva, este é substituído pela média de seus pontos adjacentes.

Para a substituição de faixas do sinal, são aplicadas técnicas de regressão baseadas no princípio dos mínimos quadrados resultando na obtenção dos coeficientes da equação da reta, parábola ou cúbica, através da solução dos seguintes sistemas de equações lineares [Dorn, 79],

a) para a reta:

$$y = a_0 + a_1x; \quad (4.5)$$

$$\begin{aligned} na_0 + (\sum x)a_1 &= \sum y; \\ (\sum x)a_0 + (\sum x^2)a_1 &= \sum xy. \end{aligned}$$

b) para a parábola:

$$y = a_0 + a_1x + a_2x^2; \quad (4.6)$$

$$\begin{aligned} na_0 + (\sum x)a_1 + (\sum x^2)a_2 &= \sum y; \\ (\sum x)a_0 + (\sum x^2)a_1 + (\sum x^3)a_2 &= \sum xy; \\ (\sum x^2)a_0 + (\sum x^3)a_1 + (\sum x^4)a_2 &= \sum x^2y. \end{aligned}$$

c) para a cúbica:

$$y = a_0 + a_1x + a_2x^2 + a_3x^3; \quad (4.7)$$

$$\begin{aligned} ma_0 + (\sum x) a_1 + (\sum x^2) a_2 + (\sum x^3) a_3 &= \sum y; \\ (\sum x) a_0 + (\sum x^2) a_1 + (\sum x^3) a_2 + (\sum x^4) a_3 &= \sum xy; \\ (\sum x^2) a_0 + (\sum x^3) a_1 + (\sum x^4) a_2 + (\sum x^5) a_3 &= \sum x^2y; \\ (\sum x^3) a_0 + (\sum x^4) a_1 + (\sum x^5) a_2 + (\sum x^6) a_3 &= \sum x^3y, \end{aligned}$$

onde m é o número de amostras.

Na obtenção dos parâmetros das equações (4.5), (4.6) e (4.7), destaca-se que as curvas a serem obtidas referem-se ao gráfico plotado, tomando-se "x" como a abcissa, que corresponde ao tempo, e "y" a ordenada, ou seja, o valor do sinal no ponto em questão.

Com a equação da curva, são obtidos os novos valores do sinal na faixa selecionada, através da substituição dos correspondentes valores de "x".

Quando o tratamento da curva corresponder à parte em que o sistema está em regime permanente, o usuário pode estender a faixa a ser tratada, do primeiro ponto selecionado até o fim do arquivo do sinal, embora estes dados não estejam plotados no gráfico. Neste caso, determinando-se uma linearização, os pontos selecionados são substituídos pelo seu valor médio, garantindo-se que a resposta do sistema apresente um valor constante em regime permanente.

Caso o resultado de alguma regressão seja considerado inadequado, a rotina dispõe de uma função que recupera a curva original. Esta função, também é definida sobre uma faixa do gráfico selecionada pelo usuário, possibilitando-se que outros tratamentos não sejam alterados.

Um aspecto que pode influenciar o procedimento de regressão é o número de pontos que estão envolvidos em cada faixa selecionada, sendo

desejável que o período de amostragem do sinal, permita uma boa visualização de toda a sua parte transitória.

Todo o procedimento é realizado de forma interativa e com a participação direta do usuário na seleção das faixas a serem tratadas e na escolha do tipo de regressão a ser realizada. A Fig. 4.9 ilustra os resultados obtidos com a utilização da rotina de tratamento de sinais no tempo, sendo apresentados os gráficos de um sinal antes e após o tratamento.

4.6.2 ANÁLISE E TRATAMENTO NA FREQUÊNCIA

Uma segunda alternativa para o tratamento de sinais implementada no programa, consiste na aplicação de filtros digitais, que são projetados segundo as características do ruído e da resposta em frequência do sistema.

As análises dos espectros em frequência dos sinais de entrada e saída de um sistema nos dão uma noção da sua resposta em frequência em relação ao sinal de excitação utilizado. Desta forma, podemos impedir que a utilização de um filtro inadequado venha a modificar sensivelmente as características da resposta do sistema e conseqüentemente os resultados da identificação.

A ferramenta que permite a determinação do espectro em frequência de um sinal é a transformada de Fourier. No caso discreto, esta exige uma grande quantidade de cálculos, e conseqüentemente, um tempo de processamento elevado. Para solucionar este problema diversos autores propuseram algoritmos alternativos para obtenção da transformada de Fourier discreta (TFD) com reduzida complexidade computacional em relação à sua realização original [Bergland, 69]. Tais algoritmos são denominados por FFT - Fast Fourier Transform (Transformada Rápida de Fourier). Assim, foi utilizado (adaptado e incorporado ao programa) para a obtenção do espectro em frequência dos sinais em questão, um algoritmo para a FFT - dezimação no tempo - radix 2, cujo programa que o

implementa foi desenvolvido no LINSE - Laboratório de Instrumentação Eletrônica do Departamento de Engenharia Elétrica da UFSC.

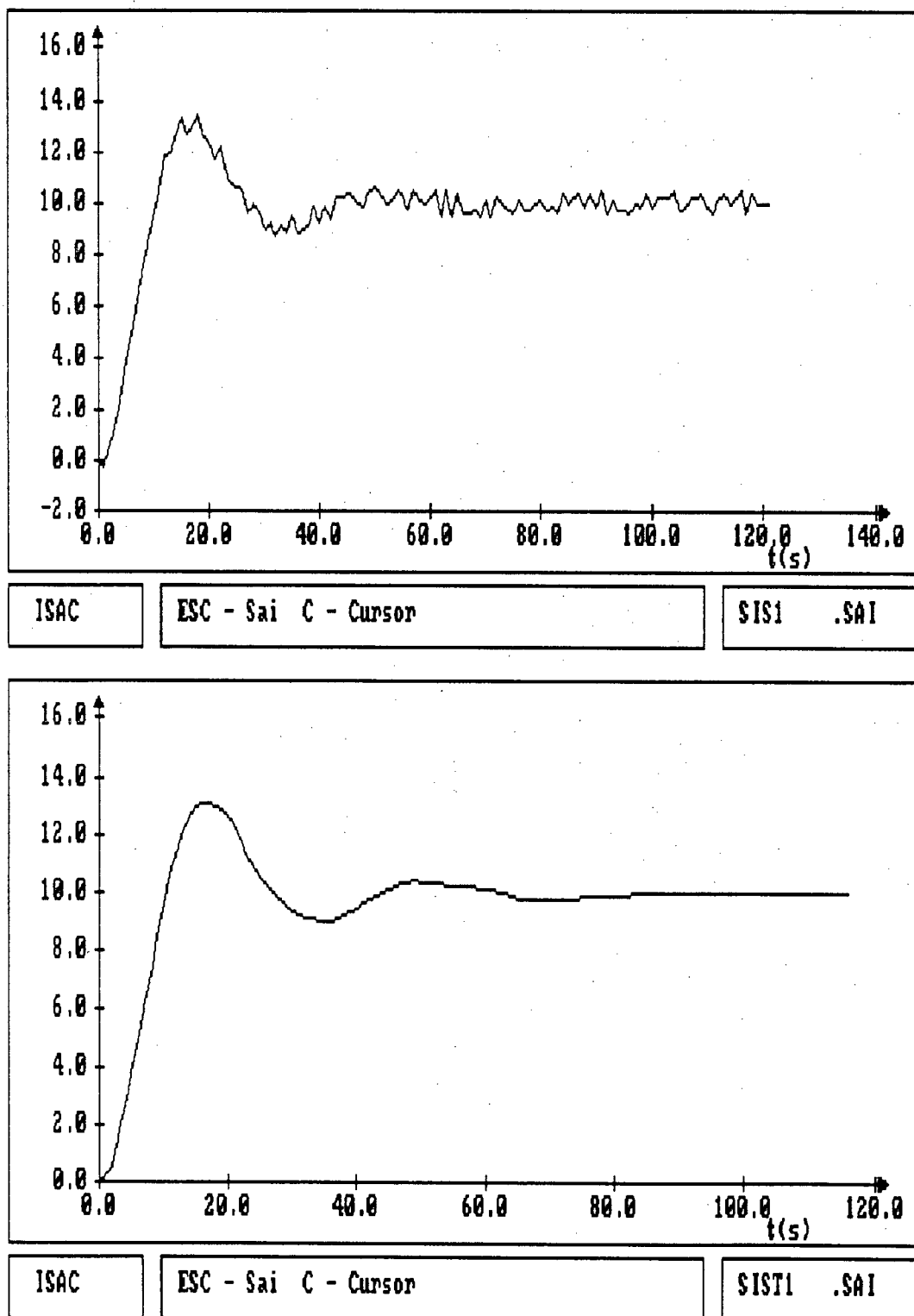


Fig. 4.9 - Resultados obtidos com o procedimento de tratamento de sinais no tempo.

Em relação ao tipo de filtro implementado, devemos ressaltar que não faz parte dos objetivos deste trabalho, o projeto de filtros, mas sim a sua utilização no tratamento de sinais. Assim sendo, optou-se pela utilização de rotinas de pouca complexidade adaptadas de trabalhos desenvolvidos dentro da área.

Uma das abordagens de grande utilização em projeto de filtros digitais, devido à simplicidade de realização, e que será empregada neste trabalho, consiste na sua obtenção a partir de filtros analógicos.

Inicialmente, optou-se pela utilização de um filtro passa baixas, que permite a passagem das componentes de baixas frequências do sinal, com uma pequena perda, e atenua as componentes de altas frequências, sendo caracterizado pelos seguintes parâmetros:

ω_p - frequência de corte de banda passante;
 ω_s - frequência de corte de banda de rejeição;
 A_{\min} - atenuação mínima na banda de rejeição;
 A_{\max} - atenuação máxima na banda passante,

ilustrados pela Fig. 4.10.

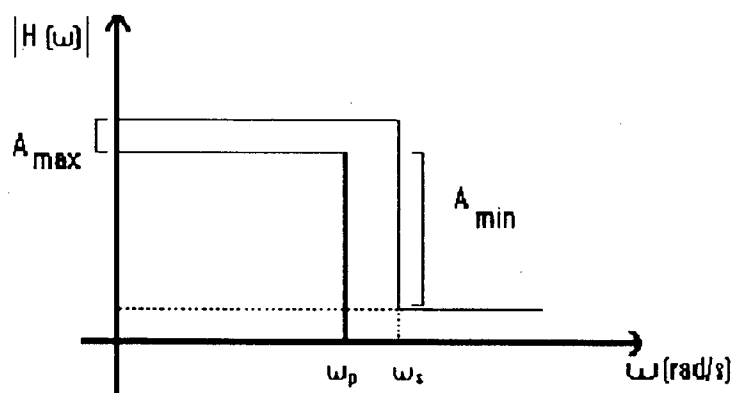


Fig. 4.10 - Característica do gabarito de um filtro passa baixas

No projeto do filtro passa baixas digital, utiliza-se como filtro analógico intermediário um filtro tipo Butterworth, que é calculado em função das frequências de corte e atenuações especificadas [Daryanani, 76].

Dentre as técnicas utilizadas para a obtenção do filtro digital a partir do filtro analógico, optou-se pela utilização da transformação bilinear, para a qual

$$S = \frac{2}{T} \frac{1-z^{-1}}{1+z^{-1}} \quad (4.8)$$

onde "T" é o período de amostragem.

Um aspecto que deve ser considerado quando do projeto do filtro, é que nem sempre a presença do ruído é detectada em uma faixa específica de frequência, podendo este estar somado a todas as componentes. Nestes casos, a especificação do filtro de forma a se eliminar as componentes do ruído que apresentem-se isoladas, deve atenuar a sua influência sobre o processo de identificação. De qualquer forma, a sua aplicação implica na modificação, em parte, das características do sinal, devendo ser utilizado tanto para o sinal de entrada como para o sinal de saída, de forma a se manter aproximadamente inalterado o comportamento entrada-saída do sistema.

O desenvolvimento deste módulo foi realizado considerando-se a utilização de uma interface de alto nível que venha a auxiliar ao usuário na especificação das características do filtro, já que não se pode requerer deste um completo conhecimento da área, uma vez que o projeto de filtros não é a finalidade principal do programa. Para tanto, a definição do filtro é realizada com o auxílio de um esquema o qual traduz o significado dos parâmetros que deverão ser fornecidos pelo usuário.

A implementação das rotinas de cálculo dos parâmetros dos filtros, foram realizadas com a adaptação à estrutura do programa, de rotinas desenvolvidas no pacote computacional PAIR - Programa para Projeto e Análise de Filtros IIR, desenvolvido no LINSE/EEL/UFSC.

4.7 A BASE DE MÉTODOS DE IDENTIFICAÇÃO

Conforme a especificação realizada no capítulo anterior, para a base de métodos do programa implementamos diversos algoritmos baseados em três diferentes abordagens: no princípio dos mínimos quadrados; nas características de reposta ao degrau e na correlação entre os sinais de entrada e saída do sistema.

4.7.1 ALGORITMOS BASEADOS NO PRINCÍPIO DOS MÍNIMOS QUADRADOS

O princípio dos mínimos quadrados foi originalmente desenvolvida por Gauss no começo do Século XIX, sendo formulada como: "Os valores mais prováveis para quantidades desconhecidas são aqueles que minimizam a soma dos quadrados das diferenças entre valores medidos e calculados com a utilização desses valores" [Goodwin, 77]. Desta forma, dentro do contexto de equivalência definido no Capítulo 2, obtém-se que o critério a ser minimizado para a determinação dos parâmetros do modelo é a soma dos quadrados das diferenças entre a saída real do sistema e a obtida com a simulação do modelo identificado.

Na base de métodos do programa, foram implementados quatro algoritmos que se utilizam deste critério para a identificação: o método dos mínimos quadrados simples; o método dos mínimos quadrados generalizados; o método das variáveis instrumentais e o método da matriz estendida.

Na aplicação destes algoritmos, devem ser fornecidas, além dos sinais de entrada e saída do sistema, as informações sobre a estrutura do modelo que se deseja identificar e, opcionalmente, condições iniciais para o mesmo. Para a entrada de parâmetros iniciais, são utilizados arquivos virtuais de modelos identificados por outros métodos ou previamente editados pelo usuário.

4.7.1.1 Mínimos Quadrados Simples

O algoritmo dos mínimos quadrados simples, apesar de apresentar um desempenho tendencioso na presença de ruídos, tem a vantagem de uma implementação de baixo custo computacional, sendo os modelos por ele identificados utilizados como parâmetros iniciais para outros métodos [Isermann, 80].

Seja um sistema linear descrito pela equação (4.4), que pode ser reescrita na forma vetorial como:

$$y(k) = \underline{x}^t(k) \underline{A} \quad (4.9)$$

onde: $\underline{A}^t(k) = [a_1 \ a_2 \dots a_n \ b_0 \ b_1 \ b_2 \dots b_n] ;$
 $\underline{x}^t(k) = [-y(k-1) \dots -y(k-n) \ u(k-a) \ u(k-1-a) \dots u(k-n-a)] ;$

Considerando que o vetor "A" é desconhecido e "A'" representa a sua estimativa, a aplicação do princípio dos mínimos quadrados resulta no seguinte critério a ser minimizado:

$$S = \sum_{i=0}^{m-1} [y(k-i) - \underline{x}^t(k-i) \underline{A}']^2 \quad (4.10)$$

onde "m" é o número de amostras do experimento.

Na forma vetorial a equação (4.10) fica:

$$S = (\underline{Y} - \underline{X} \underline{A}')^t (\underline{Y} - \underline{X} \underline{A}') \quad (4.11)$$

com: $\underline{Y}^t = [y(k), y(k-1), \dots, y(k-m+1)] ;$

$$\underline{X} = \begin{bmatrix} \underline{x}^t(k) \\ \underline{x}^t(k-1) \\ \vdots \\ \underline{x}^t(k-m+1) \end{bmatrix}$$

Diferenciando-se parcialmente a equação (4.11) em relação a \underline{A}' e igualando-se a zero tem-se que o valor de \underline{A}' que minimiza S satisfaz a equação:

$$(\underline{X}^t \underline{X}) \underline{A}' = \underline{X}^t \underline{Y} \quad (4.12)$$

Se $x^t x$ é inversível, existe uma única solução que pode ser expressa por [Goodwin, 77; Strejc, 80]:

$$A' = [x^t x]^{-1} x^t y \quad (4.13)$$

A versão recursiva do método mínimos quadrados é obtida aplicando-se sobre a equação (4.13) o lema de inversão de matrizes, resultando no seguinte algoritmo [Soderstrom, 78; Goodwin, 77]:

$$\begin{aligned} A'_{k+1} &= A'_k + K_{k+1}(Y_{k+1} - x^t(k) A'_k); \\ K_{k+1} &= P_k x(k+1)^t / (1 + x^t(k+1) P_k x(k+1)); \end{aligned}$$

sendo "K" o vetor de ganho variável e "P" calculado por

$$P_{k+1} = P_k (I - P_k (x(k+1) x^t(k+1) / (1 + x^t(k+1) P_k x(k+1))));$$

Encontra-se na literatura [Isermann, 74; Saridis, 74; Strejc, 78], para este mesmo algoritmo, diversas modificações que podem ser realizadas em função da estrutura escolhida para o modelo, sendo consideradas, por exemplo, ordens distintas para o numerador e o denominador.

4.7.1.2 Mínimos Quadrados Generalizados

Em função da presença de ruídos nos sinais do sistema, o algoritmo dos mínimos quadrados generalizados considera na sua estrutura um filtro digital que é igualmente identificado pelo método. Desta forma, a equação (4.4) é modificada, sendo a ela acrescida uma variável de erro [Isermann, 74; Soderstrom, 78; Strejc, 80; Goodwin, 77].

$$y(k) = b_0 u(k-a) + \frac{b_1 u(k-1-a)}{c_1 e^{k-1}} + \dots + \frac{b_n u(k-n-a)}{c_n e^{k-n}} - \frac{a_1 y(k-1)}{c_1 e^{k-1}} - \dots - \frac{a_n y(k-n)}{c_n e^{k-n}} \quad (4.14)$$

A identificação dos parâmetros do filtro é obtida, utilizando-se a mesma versão recursiva do algoritmo mínimos quadrados simples, sobre os vetores A_1' e $x_1(k)$ constituídos por:

$$\underline{A1}^t(k) = [c_1 \ c_2 \ \dots \ c_n] \\ \underline{x1}(k) = [-e(k-1) \ \dots \ -e(k-n)],$$

onde $e(k)$ é o erro entre a saída real do sistema e a saída estimada do modelo no instante k .

A cada passo do algoritmo são identificados os parâmetros do modelo e do filtro que é aplicado sobre os sinais a serem utilizados na próxima iteração, objetivando minimizar a presença de ruídos nestes sinais.

O algoritmo de mínimos quadrados generalizados tem um bom desempenho em relação a ruídos brancos, entretanto, estimações tendenciosas podem ser obtidas. O seu custo computacional é elevado, pois duas estruturas de identificação devem ser utilizadas, elevando conseqüentemente o tempo de processamento [Isermann, 80].

4.7.1.3 Variáveis Instrumentais

O algoritmo das variáveis instrumentais, considera a utilização de um sinal extra, obtido pelo método, e que está correlacionado somente com os sinais do sistema, desprezando a parcela de ruído. O método apresenta uma boa performance para vários tipos de ruídos, sendo o seu custo computacional considerado médio em comparação aos demais algoritmos [Isermann, 80].

A versão recursiva do método é obtida acrescentando-se ao algoritmo dos mínimos quadrados simples um vetor com as variáveis instrumentais x' [Soderstrom, 78; Strejc, 80; Goodwin, 77], constituído por

$$x'^t(k) = [-y'(k-1) \ \dots \ -y'(k-m) \ u(k-a) \ u(k-1-a) \ \dots \ u(k-m-a)]$$

sendo $y'(k)$ dado por

$$y'(k) = \underline{x}'(k) \underline{A}'^t(k),$$

e resultando no seguinte algoritmo:

$$\begin{aligned}\underline{A}'_{k+1} &= \underline{A}'_k + \underline{K}_{k+1}(\underline{Y}_{k+1} - \underline{x}^t(k)\underline{A}'_k); \\ \underline{K}_{k+1} &= \underline{P}_k \underline{x}'(k+1) / (1 + \underline{x}^t(k+1)\underline{P}_k \underline{x}'(k+1)); \\ \underline{P}_{k+1} &= \underline{P}_k (\underline{I} - \underline{P}_k (\underline{x}'(k+1)\underline{x}^t(k+1) / (1 + \underline{x}^t(k+1)\underline{P}_k \underline{x}'(k+1))))).\end{aligned}$$

4.7.1.4 Método da Matriz Estendida

É considerado uma aproximação da versão recursiva do método da máxima verossimilhança, com um custo computacional elevado e apresentando resultados satisfatórios na presença de ruídos [Soderstrom, 78; Goodwin, 77; Åström, 80].

Em relação à sua implementação, o método da matriz estendida também considera a presença de ruídos sobre o sinal de saída do sistema, o qual gera um erro de estimação, conforme a equação (4.14), que é calculado como no método dos mínimos quadrados generalizados, sendo a estimação deste erro acrescentada aos vetores \underline{A}' e \underline{x} .

$$\frac{\underline{A}'}{\underline{x}}^t(k) = \begin{bmatrix} a_1 & a_2 & \dots & a_n & b_0 & b_1 & b_2 & \dots & b_n & c_1 & c_2 & \dots & c_m \\ -\underline{y}(k-1) & \dots & -\underline{y}(k-n) & 1 & \underline{u}(k-a) & \dots & \underline{u}(k-n-a) & \dots & \underline{e}(k-1) & \dots & \underline{e}(k-n) \end{bmatrix}$$

A estrutura do algoritmo é idêntica à versão recursiva do método dos mínimos quadrados simples.

4.7.2 MÉTODO DA CORRELAÇÃO

No método da correlação, a identificação de um modelo linear é realizada em duas etapas. Na primeira, é obtida a seqüência ponderada, um modelo não paramétrico correspondendo a versão discreta da resposta ao impulso. A seqüência ponderada é calculada em função da correlação entre os sinais de entrada e saída do sistema [Cardoso F., 79; Isermann, 74; Godfrey, 80; Sage, 71].

A obtenção da seqüência ponderada é dada pela equação (4.15),

$$h(k) = R_{uy}(k) / R_{uu}(0), \quad (4.15)$$

onde $h(k)$ corresponde ao valor estimado da resposta ao impulso do sistema no tempo kt , $R_{uu}(0)$ a autocorrelação do sinal de entrada com um deslocamento nulo e $R_{uy}(k)$ a intercorrelação entre os sinais de entrada e saída deslocados de k amostras.

Em termos práticos, verificamos que o algoritmo apresenta uma implementação muito simples, e o seu desempenho está diretamente relacionado ao número de amostras utilizadas, apresentando em contrapartida um custo maior de processamento.

Na aplicação do método, é desejável a utilização da seqüência binária pseudo-aleatória como sinal de excitação, pois para garantir a obtenção de uma seqüência ponderada de boa qualidade deve-se assegurar a condição de excitação persistente.

O modelo paramétrico, é obtido através da relação linear que pode-se verificar, entre os parâmetros a_i e b_i da equação 4.16, e a seqüência ponderada do sistema [Isermann, 74; Martinez, 70], que é definida, como a resposta à entrada : $u(0)=1$; $u(1)=0$; $u(2)=0$; ... $u(m)=0$.

$$y(k) = b_0u(k-a) + b_1u(k-1-a) + \dots + b_nu(k-n-a) - a_1y(k-1) - \dots - a_ny(k-n), \quad (4.16)$$

A equação 4.16 pode então ser reescrita como:

$$h(k) = b_0u(k-a) + b_1u(k-1-a) + \dots + b_nu(k-n-a) - a_1h(k-1) - \dots - a_nh(k-n) \quad (4.17)$$

Variando k de 1 até $2n+1$, na equação acima, obtém-se $2n+1$ equações relacionando à seqüência $h(k)$ e os parâmetros do modelo, que são obtidos, através da resolução destas equações.

4.7.3 MÉTODOS GRÁFICOS

Na implementação dos métodos gráficos, são utilizadas as características de resposta ao degrau do sistema para a estimação dos parâmetros de um modelo contínuo e pré-definido [Graupe, 76; Rake, 80].

Embora os métodos sejam referentes a modelos de pouca complexidade, seu desempenho pode ser considerado satisfatório, ao menos como primeira aproximação de identificação, uma vez que permitem determinar: constantes de tempo dominantes, tempo de subida, características do ruído existente, em fim, dados úteis para a continuidade do processo.

A presença de ruído no sinal de saída do sistema, pode dificultar a identificação dos parâmetros por estes métodos. Entretanto, os mesmos podem ser obtidos com boa aproximação após a aplicação de rotinas de análise e tratamento de sinais.

4.7.3.1. Modelos de Segunda Ordem

Em relação a modelos de segunda ordem, duas abordagens são consideradas. Inicialmente, para modelos de segunda ordem periódicos, ou seja, que apresentem oscilações sobre o valor de regime permanente antes de se estabilizarem, descritos pela equação (4.18), a identificação do atraso é realizada diretamente pelo usuário sobre o gráfico da resposta do sistema. O ganho "k" é dado pelo quociente entre as variações da saída pela entrada em regime permanente. Os valores de ω_0 e ξ são obtidos em função da medida do seu período de oscilação e do instante em que ocorre o valor máximo do primeiro sobresinal, através da solução de equações algébricas simples.

$$G(S) = \frac{K e^{-at}}{T^2 S^2 + 2\xi TS + 1} \quad (4.18)$$

K = ganho;
 T = $1/\omega_0$;
 ω_0 = frequência natural;
 ξ = amortecimento;
 a = atraso de transporte.

Para modelos de segunda ordem aperiódicos descritos pela equação (4.19), a determinação dos parâmetros "a" e "k" é feita como no caso anterior. Já as constantes de tempo são obtidas com o auxílio de um

gráfico representando o $\log(y_f - y(t))$ versus t , com y_f o valor de regime permanente do sinal de saída (vide Fig. 4.11).

$$G(S) = \frac{k e^{-at}}{(S+t_1)(S+t_2)} \quad (4.19)$$

k = ganho;
 t_1, t_2 = constantes de tempo;
 a = atraso de transporte.

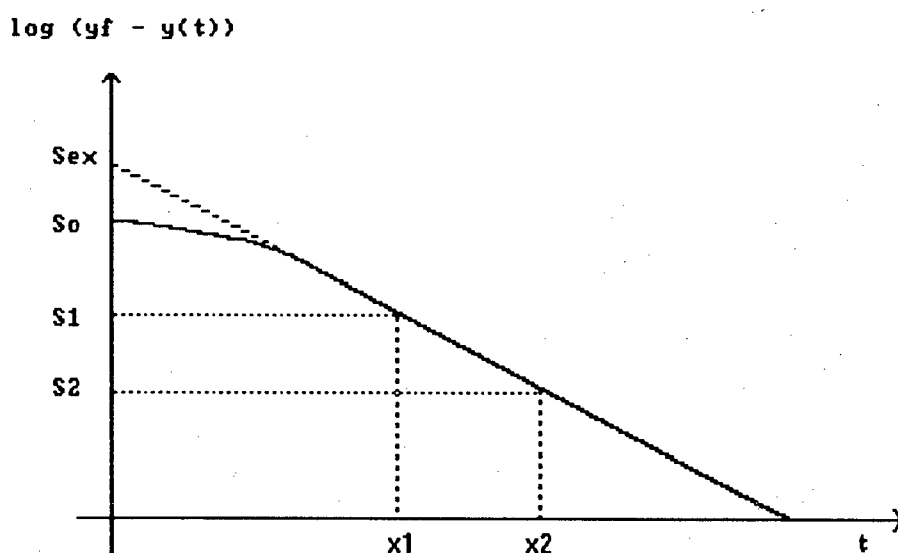


Fig. 4.11 - Método gráfico para modelos de segunda ordem aperiódicos

Sobre este gráfico são obtidos os pontos S_{ex} e S_0 , e marcados dois pontos quaisquer na parte linear da curva para a determinação da sua inclinação, medindo-se os valores de S_1 , S_2 , x_1 e x_2 . A obtenção das constantes de tempo t_1 e t_2 , é baseada na resolução das equações:

$$t_1 = \frac{0.43 \cdot (x_2 - x_1)}{\log S_1 - \log S_2} \quad (4.20)$$

$$t_1/t_2 = \frac{S_{ex}}{S_{ex} - S_0} \quad (4.21)$$

A mesma abordagem é utilizada para a determinação de modelos de primeira ordem, descritos pela equação (4.22), utilizando-se apenas a inclinação do gráfico em questão para a determinação da constante de tempo do sistema.

$$G(s) = \frac{K e^{-at}}{(TS + 1)} \quad (4.22)$$

k = ganho;
a = atraso de transporte;
T = constante de tempo.

4.7.3.2 Modelos de Alta Ordem Aperiódicos

Modelos de alta ordem para um sistema, podem ser obtidos considerando-se o posicionamento de todos os seus pólos sobre um mesmo ponto, e utilizando-se um método que é baseado na posição do ponto de inflexão da curva e nos dados obtidos de uma reta tangente a ela, passando por este ponto, (vide Fig. 4.12). Com estes dados (valores de T_a , T_b , T_c , T_d , T_e e nK) e o auxílio de tabelas, pode-se estimar a ordem e os parâmetros da equação (4.23).

$$G(s) = \frac{k e^{-at}}{(ts + 1)^n} \quad (4.23)$$

4.8 VALIDAÇÃO DE MODELOS

Para a avaliação do desempenho do modelo, em relação ao comportamento do sistema, o módulo de validação do programa, permite a sua simulação com o sinal de entrada utilizado na identificação, e a comparação do resultado com a saída real do sistema. A simulação do modelo é realizada discretamente, como no módulo de simulação. A comparação do resultado da simulação com o sinal de saída real do sistema é realizada amostra a amostra, sendo fornecidos ao usuário, o erro médio e o erro médio quadrático obtidos das diferenças entre as saídas. É possível,

ainda, a visualização gráfica dos dois sinal superpostos para a avaliação dos resultados, como ilustra a Fig. 4.13.

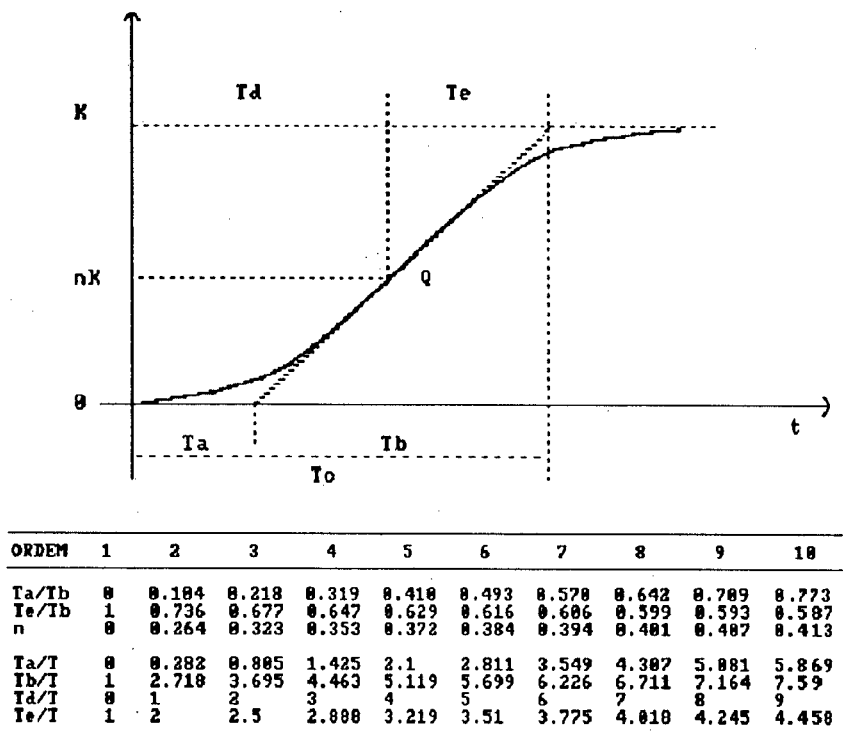


Fig. 4.12 - Método gráfico para modelos de alta ordem aperiódico

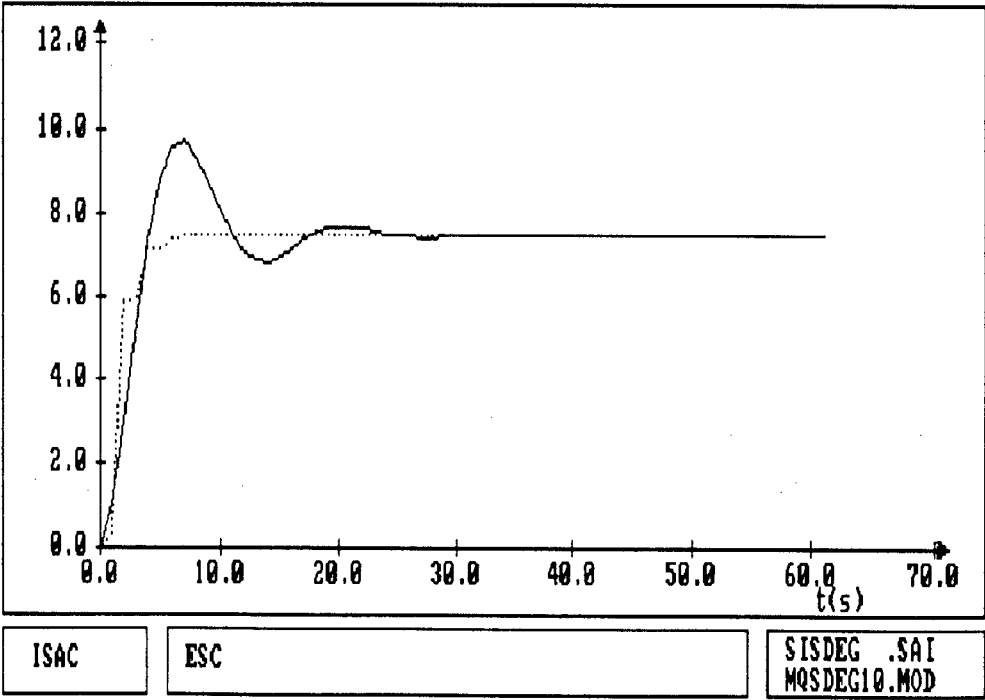


Fig. 4.13 - Gráfico gerado pela rotina de Validação

Na aplicação de algoritmos de identificação paramétricos, pressupõe-se o conhecimento da estrutura do modelo a ser identificado, o que para o caso monovariável, corresponde ao atraso e à ordem do sistema. Para a especificação da estrutura do modelo, devem ser utilizadas todas as informações prévias que se dispõe sobre o sistema, obtidas através da abordagem "modelos de conhecimento", ou da análise de modelos não paramétricos, como a resposta ao degrau e a resposta ao impulso. Entretanto, a ordem do sistema não é facilmente determinada através destes procedimentos, sendo necessário em muitos casos, o arbítrio do seu valor e a posterior avaliação dos modelos identificados.

Além de permitir a obtenção de modelos não paramétricos, o programa considera um algoritmo para a estimação da ordem do modelo, baseado na seqüência ponderada do sistema, e uma rotina para a avaliação dos seus pólos e zeros. Em função de problemas numéricos, e da presença de ruídos na seqüência ponderada, o algoritmo de pré-determinação da ordem do sistema pode ser tendencioso. Entretanto, em função de uma estimação inicial, fornecida por este, e aplicando-se a cada modelo identificado o algoritmo de avaliação de pólos e zeros, pode-se garantir, em poucas iterações, a obtenção da estrutura que melhor se enquadra às características do sistema sob estudo, uma vez que, verificando-se a presença de pólos e zeros coincidentes, conclui-se que o modelo foi identificado com uma estrutura acima da necessária à sua representação.

No procedimento de validação de sistemas deve ser considerado um aspecto inerente a todo o processo de identificação que pode dificultar a avaliação dos resultados. Desta forma, deve-se ter em mente que o modelo do sistema é apenas uma descrição aproximada do seu comportamento entrada-saída, sendo este diretamente relacionado às condições do experimento de identificação. Por exemplo, se o modelo de um sistema é obtido utilizando-se como excitação um sinal senoidal com uma única frequência, não poderemos garantir a utilização deste modelo para

descrever o comportamento de tal sistema em resposta a outros sinais de excitação.

4.8.1 - ALGORITMO DE PRÉ-DETERMINAÇÃO DA ORDEM DO MODELO [SINHA, 83]

Dada a seqüência ponderada do sistema, h_0, h_1, \dots, h_m , o algoritmo baseia-se na determinação do posto da matriz de Hankel, definida como:

$$H(l, k) = \begin{vmatrix} h_k & h_{k+1} & h_{k+2} & \dots & h_{k+l-1} \\ h_{k+1} & h_{k+2} & h_{k+3} & \dots & h_{k+l} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ h_{k+l-1} & h_{k+l} & h_{k+l+1} & \dots & h_{k+2l-2} \end{vmatrix} \quad (4.24)$$

, onde "l" é a ordem estimada para o modelo, e "k" varia de 0 a $m-2l+2$. Pode-se determinar a ordem do sistema, avaliando os determinantes de $H(l, k)$, para cada "k" e diferentes valores de "l", sendo que estes devem ser nulos, quando "l" for superior a ordem do sistema. Na prática, em função da influência de ruídos, os determinantes não chegam a zerar, sendo introduzido um critério para estabelecer o nível de significância. Desta forma, obtém-se a média dos determinantes para cada ordem estimada, escolhendo-se aquela que apresentar um maior valor "Dl", que é definido, como o módulo da razão entre a média dos determinantes de $H(l, k)$ e a média dos determinantes de $H(l+1, K)$.

4.9 CONCLUSÃO

No desenvolvimento do programa foram seguidas, de uma maneira geral, todas as especificações realizadas, sendo observados os aspectos da modularidade e da interface de alto nível com o usuário, os quais foram alcançados através da utilização de uma estrutura de arquivos, comuns a todas as funções do pacote, e de um ambiente de gerenciamento destes arquivos e comunicação com o usuário.

Outro aspecto que foi considerado é a questão do reaproveitamento de software, já que a utilização de rotinas implementadas em outros pacotes computacionais do LCMI e do LINSE como, por exemplo, o pacote gráfico APGRAFI e as rotinas de projeto automático de filtros do programa PAIIR, possibilitou que os esforços realizados pudessem ser direcionados as questões relacionadas diretamente com a identificação de sistemas.

Por fim, queremos ressaltar que durante o desenvolvimento do programa, por se tratar de uma primeira versão, puderam ser levantadas diversas modificações ou aperfeiçoamentos que o tornariam mais completo e otimizado, sendo muitas destas questões, levantadas na conclusão deste trabalho.

CAPÍTULO 5

RESULTADOS OBTIDOS COM A UTILIZAÇÃO DO PROGRAMA

5.1 INTRODUÇÃO

Durante o desenvolvimento do programa todas as rotinas implementadas foram validadas, considerando-se a especificação de seus requisitos funcionais, sendo também verificado o comportamento do sistema como um todo, no que diz respeito ao ambiente em que estas rotinas operam.

Neste capítulo, faremos a descrição dos resultados obtidos com a utilização do programa, sendo considerados o desempenho dos algoritmos de identificação e as vantagens da utilização de rotinas de tratamento de sinais na obtenção dos modelos do sistema.

Por considerar a utilização de todas as funções do programa, os experimentos realizados não se restringiram aos módulos de identificação e tratamento de sinais, sendo utilizadas as rotinas de geração de sinais e simulação de sistemas para a obtenção dos dados do experimento. Desta forma, pudemos verificar o desempenho do programa dentro de um contexto de aplicação que considera a sua utilização no ensino dos procedimentos de identificação de sistemas.

5.2 DESCRIÇÃO DOS EXPERIMENTOS

Na apresentação dos resultados do programa, todos os passos envolvidos no procedimento de identificação, são descritos através de um exemplo. Para tal, foi utilizado um modelo discreto de 2ª ordem (equação (5.1)), o qual foi escolhido por se tratar de um modelo utilizado em vários

trabalhos com o objetivo de comparação de diversos algoritmos [Isermann, 74].

$$G(z^{-1}) = \frac{z^{-1} + 0.5z^{-2}}{1 - 1.5z^{-1} + 0.7z^{-2}} \quad (5.1)$$

Para a realização dos experimentos, foram gerados diversos sinais de excitação utilizados na simulação dos modelos, sendo esta realizada em diferentes condições. Desta forma, vários arquivos de sinais foram criados e armazenados no diretório virtual do programa, para uma posterior identificação, onde objetivou-se demonstrar a influência do sinal de identificação, da presença de ruídos e do tratamento de sinais.

Cumpre-se ressaltar, que os experimentos foram realizados considerando-se na simulação dos sistemas a presença do sustentador de ordem zero, o que permite caracterizar a resposta de um sistema ao pulso unitário, como a versão discreta de sua resposta ao impulso.

5.3 OBTENÇÃO DE INFORMAÇÕES INICIAIS SOBRE O SISTEMA

Inicialmente, é considerado o caso em que não se dispõe de um conhecimento prévio do sistema a ser identificado, sendo utilizadas duas abordagens para a obtenção de uma primeira aproximação às características do modelo: a análise da resposta ao degrau com aplicação de métodos gráficos, e a pré-determinação da ordem do modelo com vistas à aplicação dos métodos paramétricos.

5.3.1 ANÁLISE DA RESPOSTA AO DEGRAU

Objetivando-se demonstrar as vantagens obtidas através da análise gráfica da resposta do sistema, o modelo da equação (5.1), foi simulado com um sinal de entrada igual ao degrau, considerando-se um caso ideal de operação, ou seja, sem a presença de ruídos. Com a análise da resposta do sistema ao degrau, apresentada na Fig. 5.1, pode-se identificar visualmente

se este apresenta transferência direta, atraso de transporte e o tempo aproximado que leva para atingir o regime permanente.

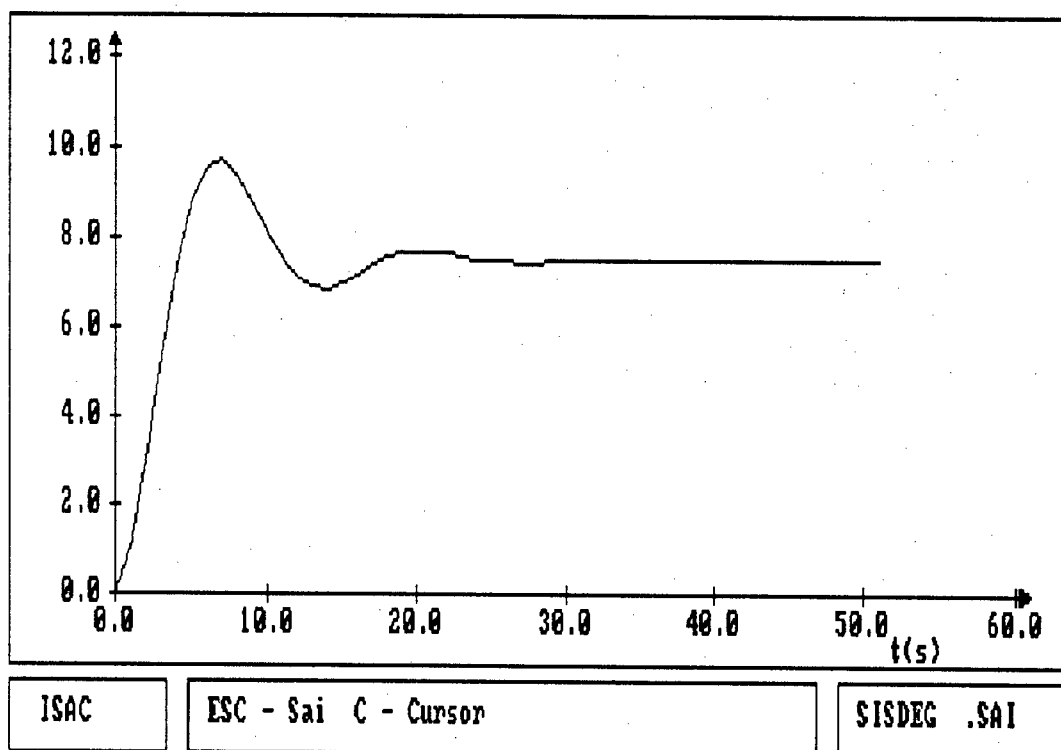


Fig. 5.1 - Resposta ao degrau do sistema

Em função da forma da curva de resposta do sistema ao degrau, optou-se pela aplicação do método gráfico de segunda ordem, sendo os resultados apresentados na Fig. 5.2. Apenas como ilustração, o modelo obtido foi discretizado e validado em relação à resposta ideal do sistema (Fig. 5.3). Em função da ausência de ruídos, pode-se concluir que os resultados foram bastantes satisfatórios, mesmo considerando-se que não se dispunha de uma boa amostra da parte transitória do gráfico, sendo o método aplicado sobre um número limitado de pontos.

Num caso real, com a presença de ruídos, o desempenho do método gráfico apresenta-se tendencioso, tornando-se necessária a aplicação de ferramentas para a análise e tratamento de sinais através de técnicas de regressão, apresentadas no capítulo anterior. De qualquer forma, os

resultados serviriam de base para a especificação de sinais de excitação mais adequados à aplicação de outros métodos de identificação.

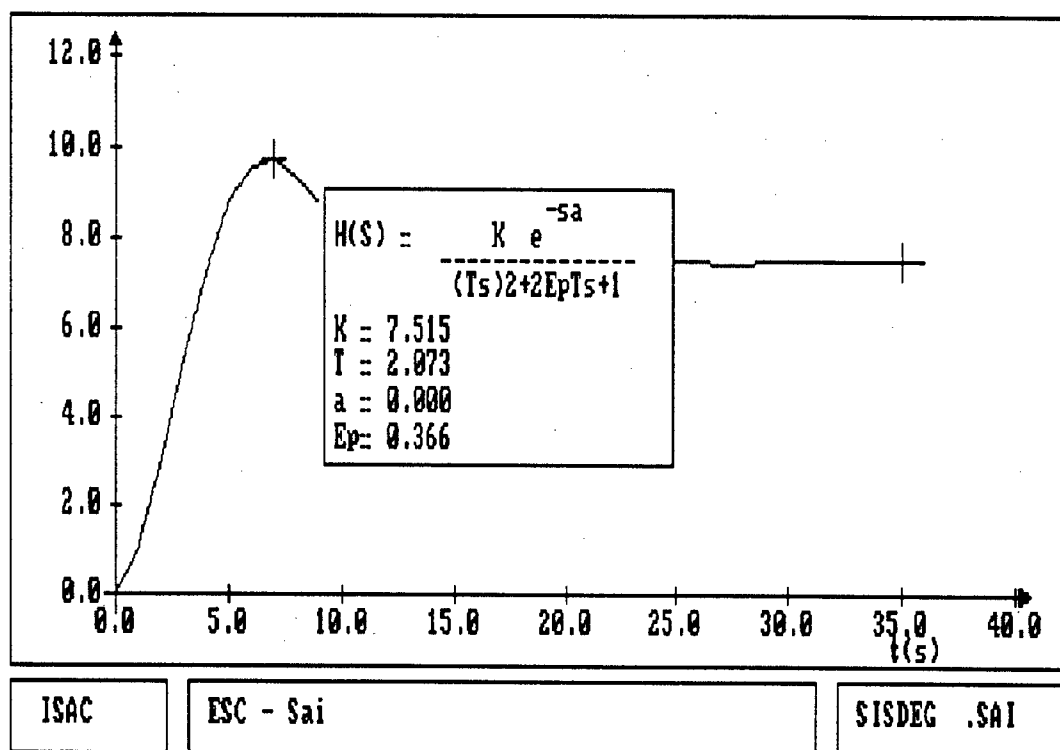
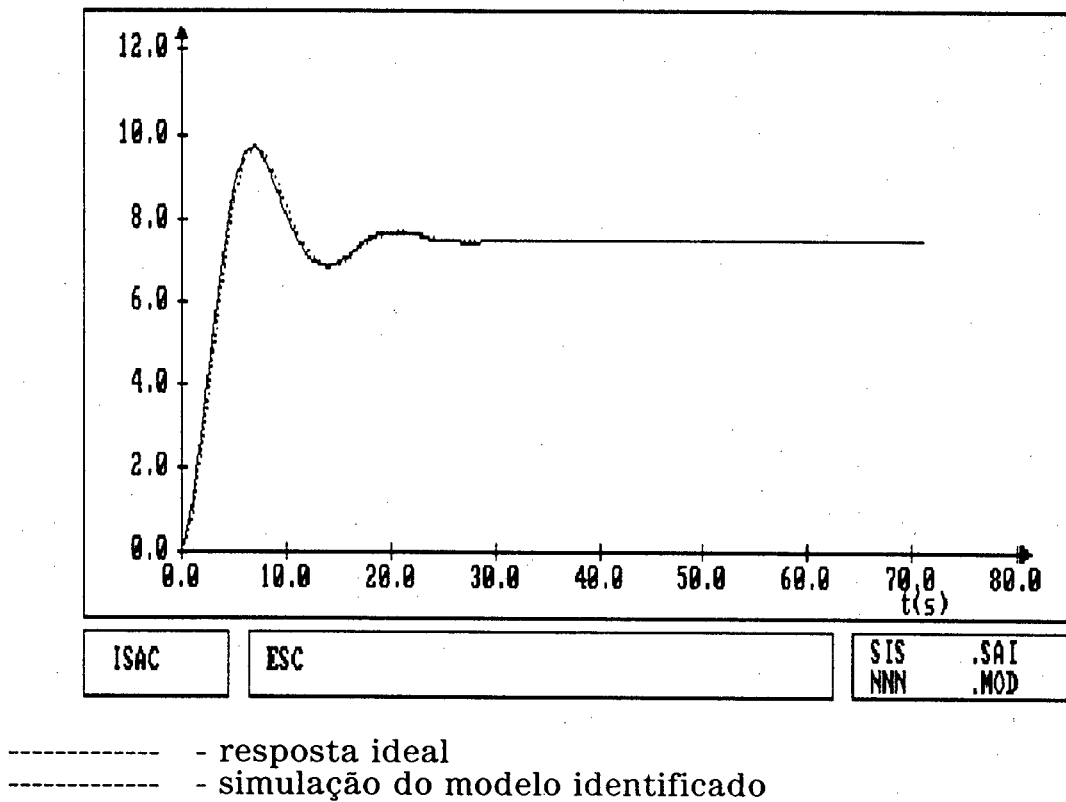


Fig. 5.2 - Resultado da aplicação do método gráfico para modelos de segundo ordem

5.3.2 PRÉ-DETERMINAÇÃO DA ORDEM DO SISTEMA

O primeiro passo para a aplicação do algoritmo de pré-determinação da ordem é a obtenção da sequência ponderada do sistema. Para tanto, duas abordagens podem ser utilizadas: a obtenção da resposta do sistema a um pulso unitário ou a aplicação do algoritmo de correlação. Objetivando ilustrar o desempenho do algoritmo em questão foi utilizada a segunda abordagem, sendo obtida a resposta do sistema a uma sequência binária pseudo-aleatória (SBPA), com ruídos de 20% sobre o valor médio do sinal. Na escolha do comprimento máximo da sequência binária é considerado o tempo que o sistema atinge aproximadamente o regime permanente, obtido na análise da resposta do sistema ao degrau, que foi de 30 segundos. Assim foi utilizada uma SBPA de comprimento máximo igual a 512. A aplicação

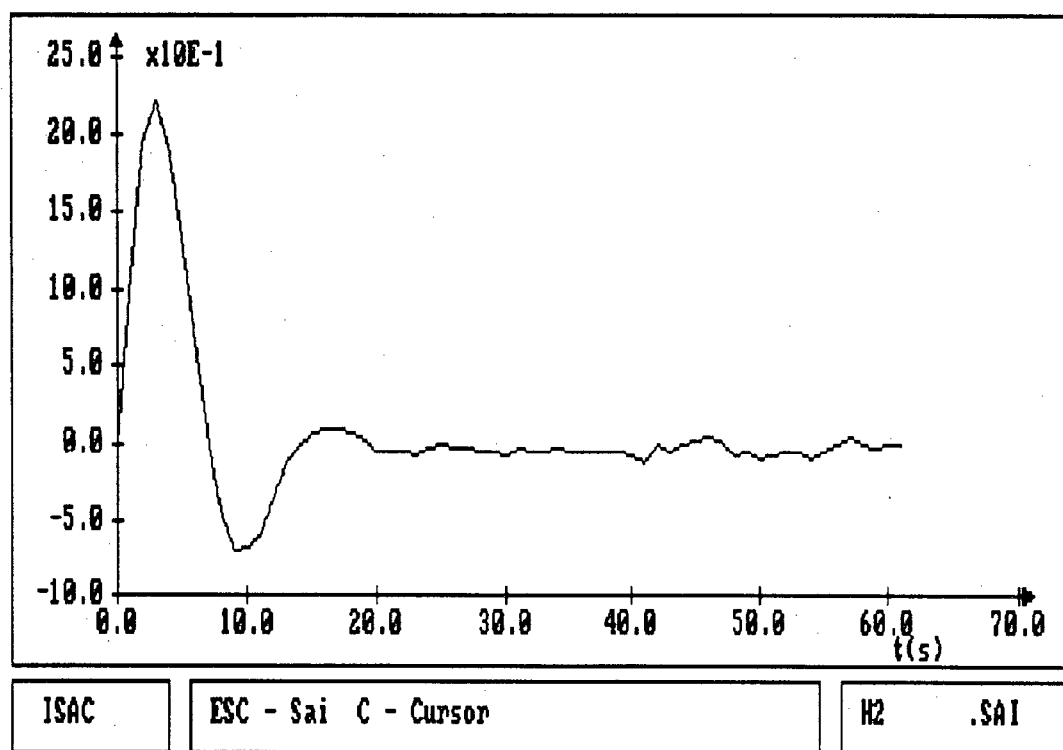
do método da correlação, resultou na seqüência ponderada apresentada na Fig. 5.4. Apenas para comparação, também é apresentada (fig. 5.4-b) a resposta do sistema a um pulso unitário sem a presença de ruídos, que no caso, corresponde à verdadeira seqüência ponderada do sistema.



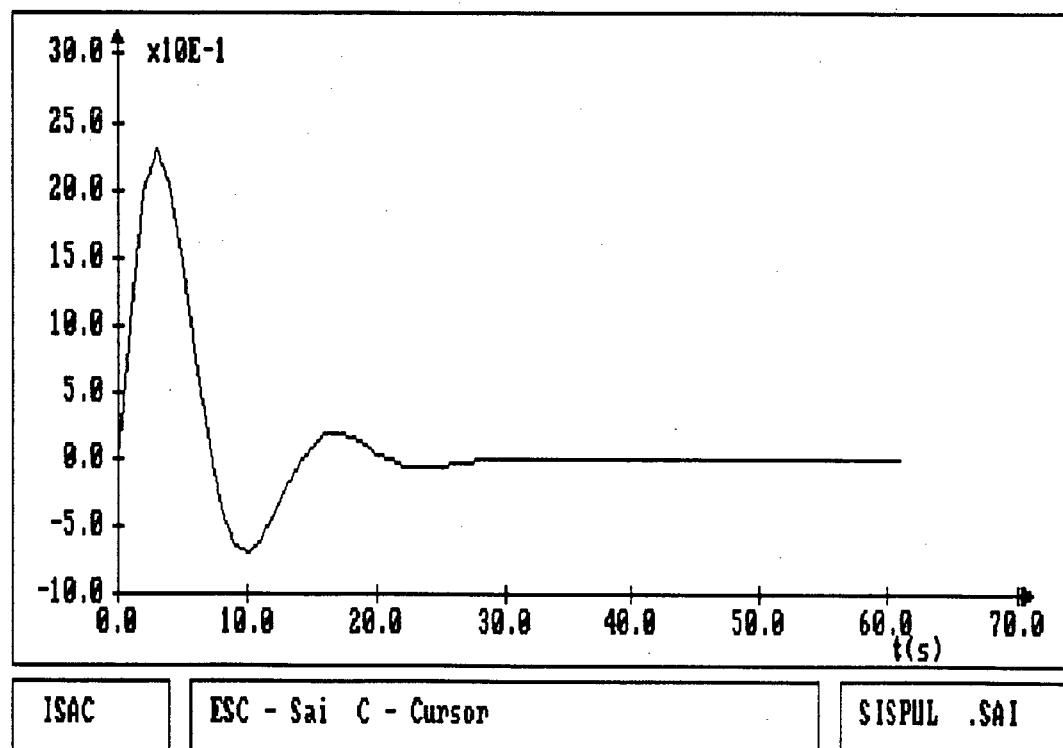
Parametros	b0	b1	b2	a1	a2
Valores Identificados	0	0.775	0.68	-1.51	0.73
Valores reais	0	1	0.5	-1.5	0.7

Fig. 5.3 - Validação do modelo identificado pelo método gráfico

Com a análise das curvas apresentadas na Fig. 5.4, pode-se concluir que o algoritmo apresentou um desempenho satisfatório, podendo este ser melhorado em função de uma ampliação no número de pontos utilizados.



a) obtida pelo método da correlação na presença de ruído



b) resposta ideal

Fig. 5.4 - Seqüência ponderada do sistema

Sobre a sequência ponderada do sistema foi aplicado o algoritmo de pré-determinação da ordem. Observa-se na Fig. 5.5, que a ordem para a qual a razão da média dos determinantes em relação a média dos determinantes de ordem imediatamente superior, em módulo, máxima, antes de decrescer, é 3.

RESULTADOS		OPCOES
Determinantes	Variacao	<input type="checkbox"/> 2 sinais <input type="checkbox"/> 1ar processos <input checked="" type="checkbox"/> 1a e 2a processos <input type="checkbox"/> ar arquivos <input type="checkbox"/> ar graficos <input type="checkbox"/> isar e tratar sinais <input type="checkbox"/> sentar Resultados
Det[1]= -8.13875888E-004	Var[1]= 2.47222800E-002	METODOS <input type="checkbox"/> Minimos quadrados Simples <input type="checkbox"/> Minimos quadrados generalizados <input type="checkbox"/> variaveis Instrumentais <input type="checkbox"/> matriz Estendida <input type="checkbox"/> Correlacao <input type="checkbox"/> metodos Graficos OPCOES <input type="checkbox"/> polos e Zeros de um modelo <input type="checkbox"/> Validar modelos <input checked="" type="checkbox"/> Pré-determinação da ordem
Det[2]= -3.29297452E-002	Var[2]= 7.79800138E+000	
Det[3]= -4.22159010E-003	Var[3]= 3.28543635E+001	
Det[4]= 1.28437090E-004	Var[4]= 1.49492821E+001	
Det[5]= 9.14616766E-006	Var[5]= 8.53612247E+001	
Det[6]= -1.07146631E-007		

Fig 5.5 - Tela do programa com o resultado do algoritmo para a pré-determinação da ordem de um sistema

Objetivando-se comprovar os resultados obtidos pelo algoritmo de pré-determinação da ordem, foi identificado um modelo de terceira ordem, pelo método dos mínimos quadrados generalizados. Após a sua validação, foi aplicado ao modelo resultante, o algoritmo para a determinação de pólos e zeros, sendo os resultados apresentados na Fig. 5.6.

Apesar da estimação do modelo ter sofrido influência de ruídos, encontramos um pólo bem próximo a um zero, e concluímos que este pode ter sido identificado com uma ordem acima da necessária para a representação do sistema.

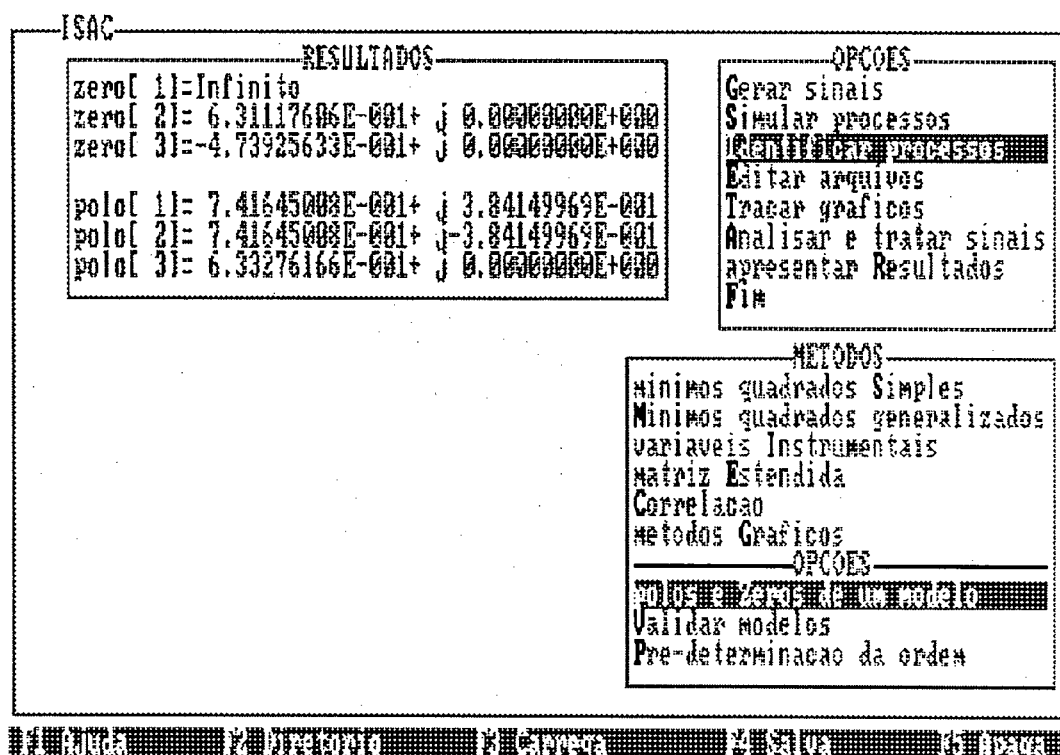


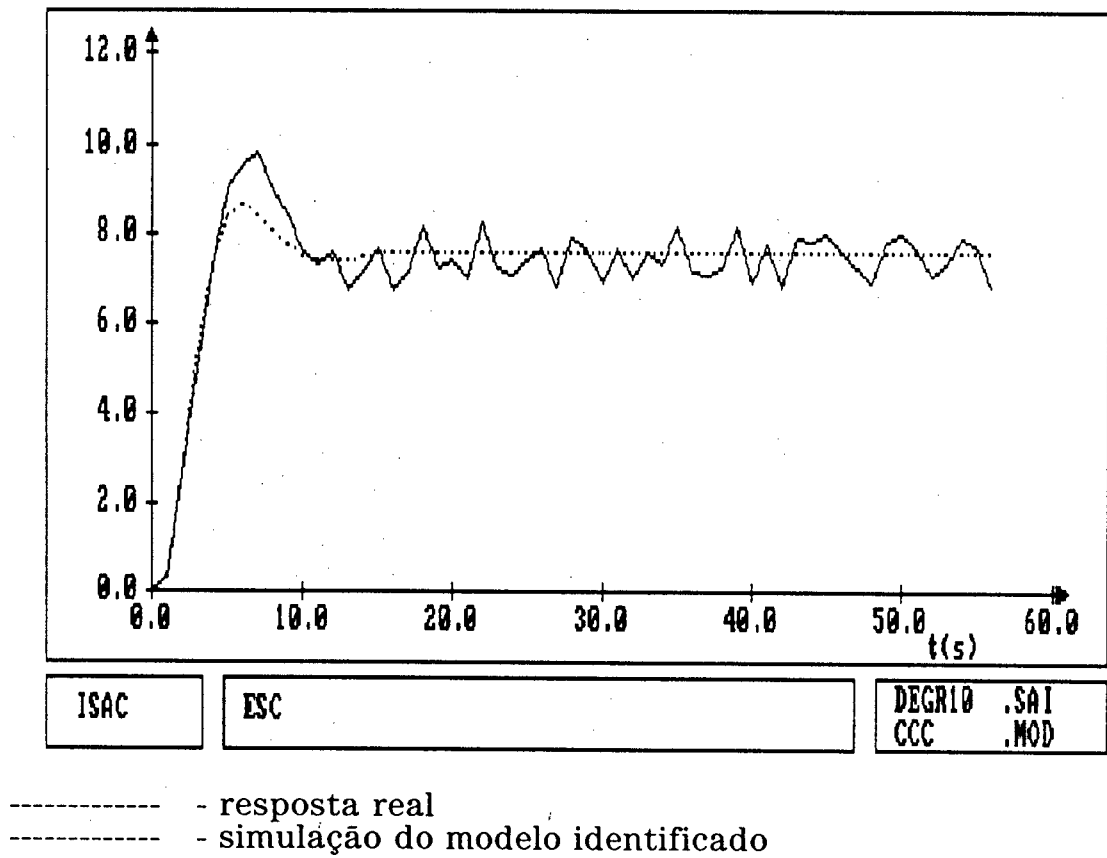
Fig 5.6 - Tela do programa com o resultado do algoritmos que determina os pólos e zeros de um modelo.

Desta forma, como já havia-se previsto no Capítulo 4, os resultados obtidos com a rotina de pré-determinação da ordem, foram influenciados pela qualidade da seqüência ponderada e por erros numéricos que podem ocorrer em virtude do número de cálculos envolvidos nesta rotina. Entretanto, a sua utilização serve como um parâmetro inicial para o procedimento de identificação de sistemas, quando não se dispõe de conhecimento suficiente sobre a estrutura do modelo.

5.4 ANÁLISE DA INFLUÊNCIA DO SINAL DE EXCITAÇÃO

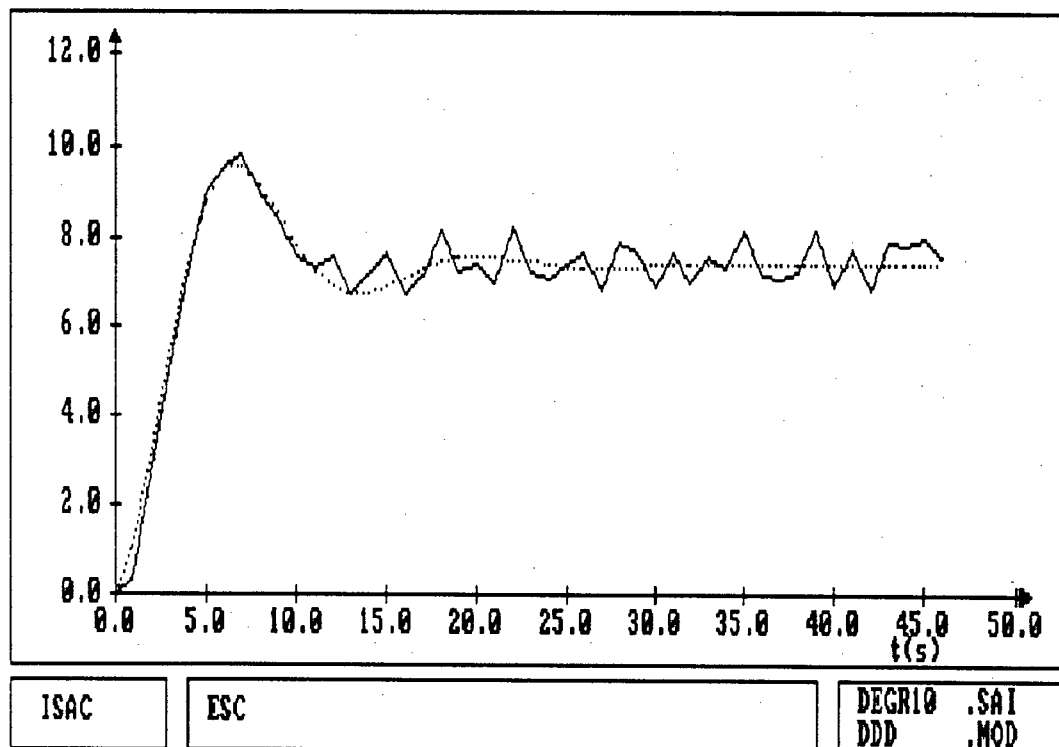
Nesta etapa, procurou-se demonstrar a influência do sinal de excitação no procedimento de identificação. Para tanto foram identificados, através do método dos mínimos quadrados generalizados, dois modelos: um utilizando-se como sinal de excitação o degrau e outro a seqüência binária pseudo-aleatória (SBPA).

Inicialmente, sem a presença de ruídos, em ambos os casos os modelos foram corretamente identificados, o que comprova a eficiência do algoritmo em relação a um caso ideal de operação. Entretanto, acrescentando-se um ruído de 10% à saída do sistema, pode-se comprovar que os modelos identificados utilizando-se como sinal de excitação uma SBPA e um degrau apresentam uma diferença significativa. As Fig. 5.7 e 5.8, apresentam os gráficos da rotina de validação para o modelo obtido com o degrau e SBPA, respectivamente. Para facilitar a visualização, em ambos os casos os modelos foram validados em relação à resposta do sistema ao degrau, com um ruído de 10%.



Parametros	b0	b1	b2	a1	a2
Valores Identificados	0	0.4002	1.8853	-1.1930	0.4958
Valores reais	0	1	0.5	-1.5	0.7

Fig. 5.7 - Resultados da identificação com sinal tipo degrau



----- - resposta real
----- - simulação do modelo identificado

Parametros	b0	b1	b2	a1	a2
Valores Identificados	0	1.0823	0.4161	-1.5006	0.7035
Valores reais	0	1	0.5	-1.5	0.7

Fig. 5.8 - Resultados da identificação com sinal tipo SBPA

A análise dos resultados, comprova que a aplicação da seqüência binária psudo-aleatória, por ser este um sinal que apresenta características de excitação persistente, resultou numa identificação mais consistente, mesmo na presença de ruídos e desconsiderando-se qualquer procedimento de tratamento de sinais.

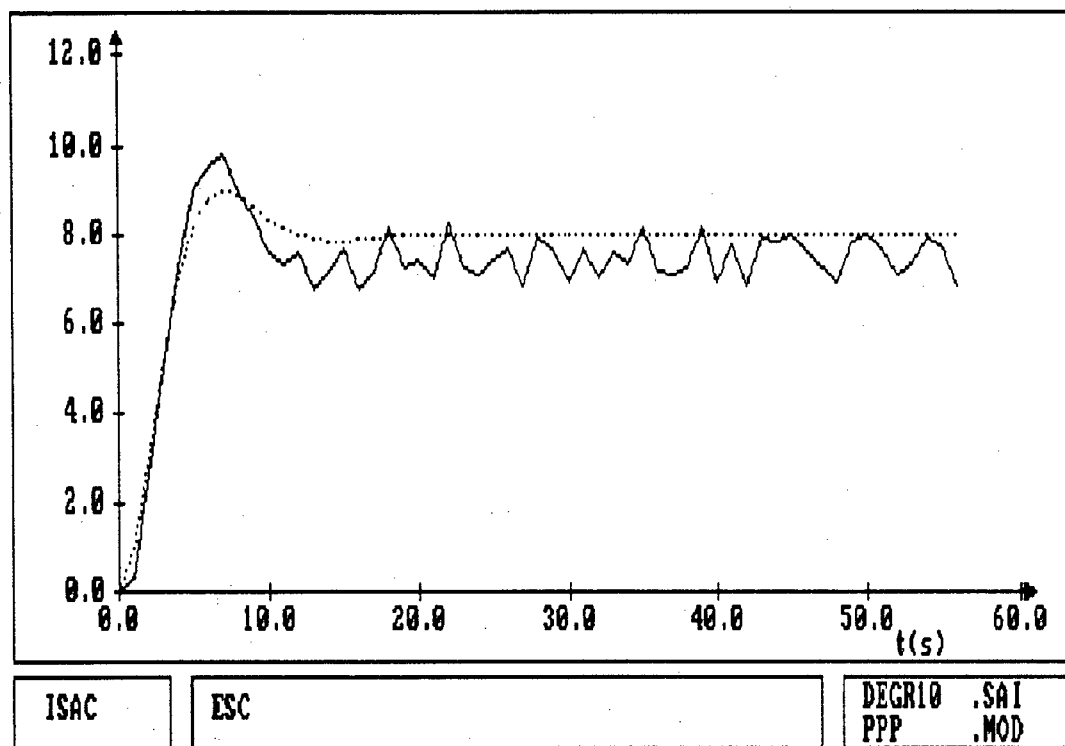
5.5 ANÁLISE DA INFLUÊNCIA DO PRÉ-TRATAMENTO DE SINAIS

Nos resultados da seção anterior, observa-se que a presença de ruídos afeta sensivelmente o procedimento de identificação de sistemas, tornando-se desejável a utilização de ferramentas para o pré-tratamento de sinais.

Para ilustrar as vantagens do pré-tratamento de sinais, considera-se inicialmente, a identificação do modelo da equação (5.1), através do método dos mínimos quadrados simples, utilizando-se um sinal de identificação tipo SBPA e acrescentando-se ao sinal de saída do sistema um ruído de 20% sobre o seu valor médio. A Fig. 5.9, apresenta os resultados da validação do modelo identificado, em relação à resposta do sistema ao degrau com ruído.

Considerando-se a qualidade dos resultados, foi realizado um pré-tratamento dos sinais de entrada e saída, para o qual utilizou-se a abordagem de tratamento de sinais na frequência. Em função do espectro em frequência dos sinais a serem filtrados, não foi verificada a atuação do ruído em frequências específicas, sendo especificado um filtro que mantivesse as características de resposta do sistema, eliminando as frequências menos significativas. O espectro em frequência dos sinais de entrada e saída, antes e após o tratamento, e os parâmetros especificados para o filtro são apresentados, respectivamente, nas Fig. 5.10, 5.11 e 5.12.

Com os sinais tratados, foi realizada uma nova identificação, utilizando-se o mesmo método, com o mesmo número de pontos e condições iniciais, verificando-se uma sensível melhora no desempenho do algoritmo, como ilustra a Fig. 5.13, onde são apresentados os parâmetros do modelo identificado e o resultado do procedimento de validação.



----- - resposta real
 ----- - simulação do modelo identificado

Parâmetros	b0	b1	b2	a1	a2
Valores Identificados	0	1.0186	0.6370	-1.3509	0.5585
Valores reais	0	1	0.5	-1.5	0.7

Fig. 5.9 - Resultado da Identificação
sem o tratamento de sinais

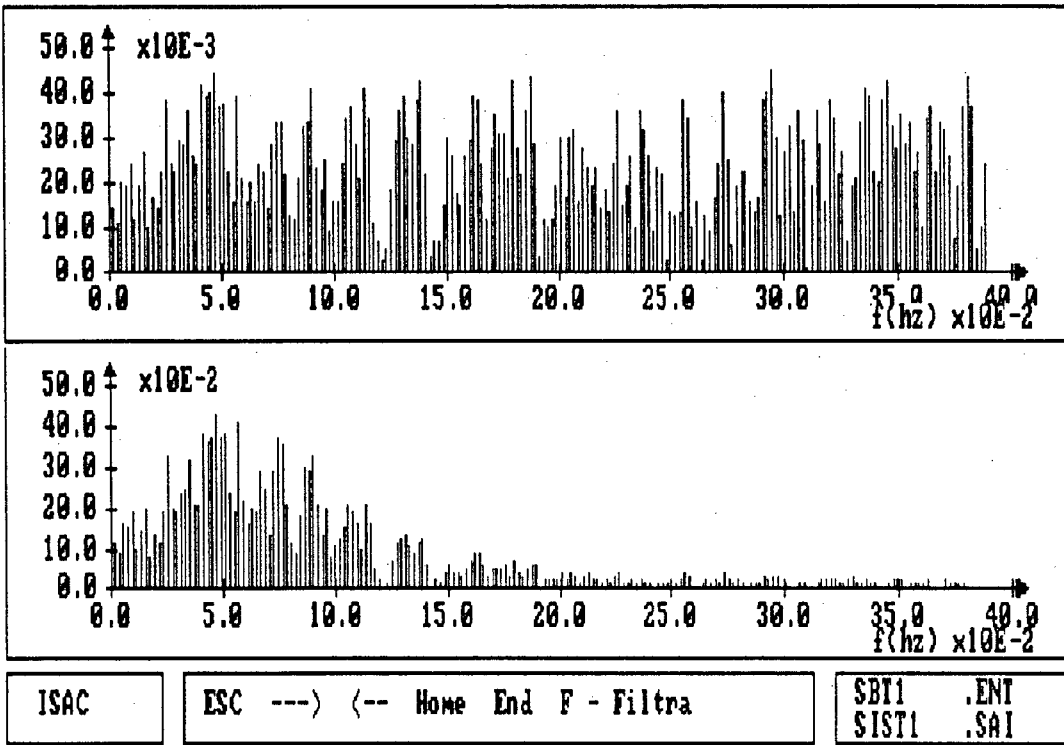


Fig. 5.10 - Espectro em frequência dos sinais antes do tratamento.

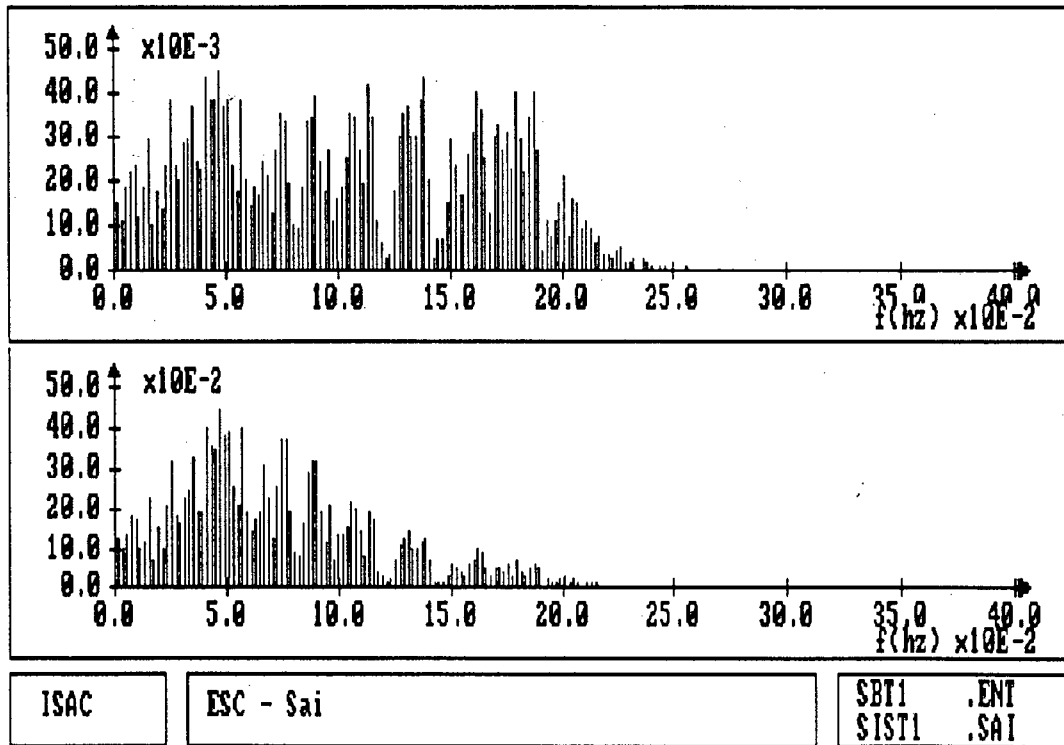


Fig. 5.11 - Espectro em frequência dos sinais após o tratamento.

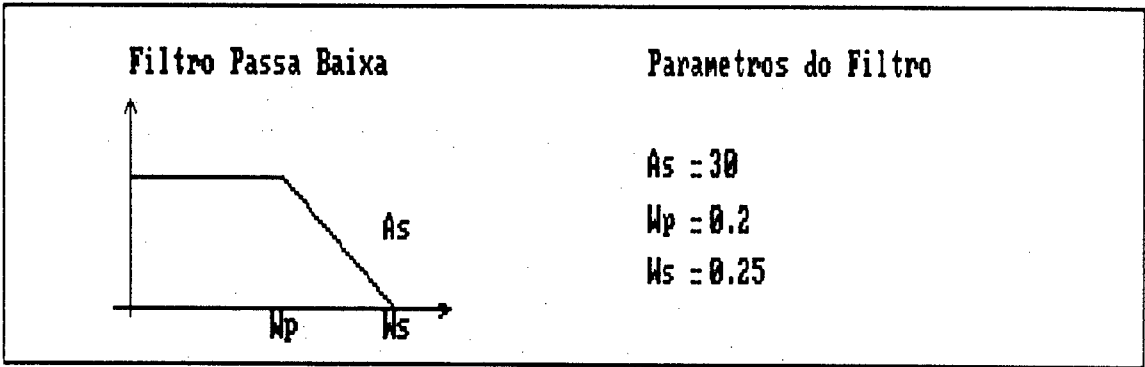
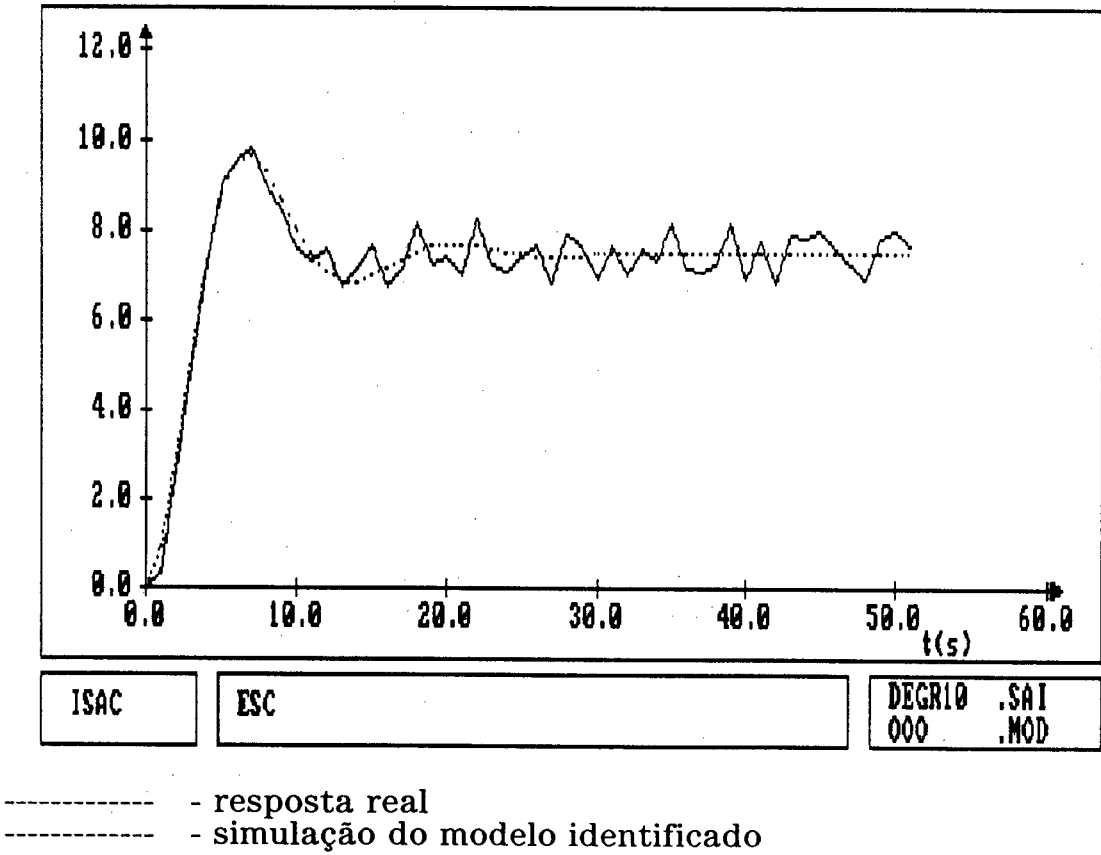


Fig. 5.12 - Parâmetros para o calculo do filtro.



Parametros	b0	b1	b2	a1	a2
Valores Identificados	0	0.9897	0.5148	-1.4965	0.6976
Valores reais	0	1	0.5	-1.5	0.7

Fig. 5.13 - Resultado da identificação
após o tratamento de sinais.

5.6 AVALIAÇÃO DO DESEMPENHO DOS ALGORITMOS

Nesta etapa, procurou-se demonstrar o desempenho dos algoritmos de identificação implementados no programa, em relação ao modelo da equação (5.1). Para tanto, foram utilizados os algoritmos dos mínimos quadrados simples, mínimos quadrados generalizados, o método das variáveis instrumentais, e o método da matriz estendida. Na aplicação dos algoritmos foram consideradas as mesmas condições de operação e número de pontos, sendo utilizado como sinal de excitação a sequência binária pseudo-aleatória, foram consideradas a presença de ruídos e o pré-tratamento de sinais. Na Tabela 5.1, são apresentados os parâmetros dos modelos identificados, incluindo-se entre estes o modelo obtido pelo método da correlação, para o qual utilizou-se um maior número de pontos e não foi usado nenhum tratamento prévio dos sinais.

Métodos:	Parâmetros					EMQ
	b0	b1	b2	a1	a2	
MQS	0	0.9897	0.5148	-1.4965	0.6976	0.0005
MQG	0	1.0046	0.6818	-1.4553	0.6711	0.0976
VI	0	0.9927	0.5091	-1.4981	0.6985	0.0001
ME	0	0.9935	0.5136	-1.4960	0.6972	0.0001
MC	0	0.9951	0.5167	-1.4851	0.6865	0.0005
Valores Reais	0	1.00	0.5	-1.5	0.7	—

MQS - Mínimos Quadrados Simples

MQG - Mínimos Quadrados Generalizados

VI - Variáveis Instrumentais

ME - Método da Matriz Estendida

MC - Método da Correlação

EMC - Erro médio quadrático

Tabela 5.1 - Resultados obtidos com a aplicação dos algoritmos de Identificação

Observando-se a Tabela 5.1, pode-se verificar através dos parâmetros identificados e do erro médio quadrático, o qual foi calculado no procedimento de validação em relação à resposta ideal do sistema ao degrau, que os algoritmos apresentaram um desempenho bastante satisfatório. Entretanto, em relação ao sistema em questão, constatamos que o desempenho do algoritmo dos mínimos quadrados generalizados foi inferior em relação ao seu próprio desempenho quando aplicado sobre sinais sem o tratamento por filtragem, onde, nas mesmas condições de operação e validação, foi obtido um erro médio quadrático de 0.0520. Desta forma, pode-se concluir que o procedimento de filtragem, em função das características especificadas para o filtro, pode beneficiar o desempenho de alguns algoritmos e prejudicar outros.

Apesar dos experimentos realizados com os algoritmos de identificação terem se limitado a um único exemplo, o que não permite a extensão dos resultados, pode-se concluir que o desempenho dos algoritmos depende diretamente do tipo de sistema sob identificação e da qualidade dos dados de entrada e saída utilizados, sendo fundamental a disponibilidade de diversos algoritmos, a comparação dos resultados e a realização de diversas iteração no tratamento de sinais, com a aplicação de filtros digitais de características distintas.

5.7 CONCLUSÃO

Com a realização dos testes descritos neste capítulo, podemos concluir que os algoritmos do programa apresentam um desempenho satisfatório em relação às suas funções dentro do procedimento de identificação, possibilitando que diversos experimentos sejam realizados dentro de um ambiente interativo.

Da mesma forma, foi verificada a importância dos aspectos relacionados à escolha do sinal de excitação e do período de amostragem e as vantagens obtidas com um tratamento prévio de sinais.

Em relação aos métodos de identificação de modelos discretos, mesmo considerando que os testes realizados não objetivaram uma completa avaliação dos mesmos, uma vez que utilizamos um modelo de pouca complexidade, podemos afirmar que os resultados obtidos confirmam as expectativas levantadas em relação ao desempenho dos mesmos.

Por fim, verificamos que, apesar de considerarem modelos de estruturas específicas, nos casos em que o sistema a ser identificado puder ser representado pelos modelos em questão, a aplicação dos métodos gráficos pode ser de grande utilidade para a identificação de sistemas.

CAPÍTULO 6

CONCLUSÃO E PERSPECTIVAS

Apresentou-se neste trabalho um pacote computacional voltado à identificação de sistemas - Identificação de Sistemas Assistida por Computador (ISAC).

Inicialmente, foram levantados os principais aspectos teóricos relacionados ao procedimento de identificação. Com base neste estudo foi especificada a estrutura do programa, destacando-se os princípios da modularidade e independência em relação à execução de suas rotinas. Desta forma, a estrutura do programa, permite que novos algoritmos e ferramentas sejam introduzidos em seu ambiente sem grandes alterações.

No desenvolvimento deste trabalho, destacamos a criação de uma base de métodos de identificação, onde podem ser utilizados algoritmos gráficos, estatísticos e baseados na correlação entre os sinais de entrada e saída de um sistema.

Um aspecto de grande importância considerado no desenvolvimento do programa, é a presença de ruídos agregados aos sinais do sistema, o que, em alguns casos, pode prejudicar sobremaneira o desempenho dos algoritmos de identificação. Assim foi desenvolvido, para aplicação nestes casos, ferramentas para o tratamento prévio de sinais, através de técnicas de regressão e utilização de filtros digitais.

O programa abrange ainda as demais etapas do procedimento de identificação, tendo sido desenvolvidas ferramentas para a geração de sinais, simulação de sistemas e validação dos modelos identificados.

Além das rotinas voltadas ao procedimento de identificação, foi desenvolvido um ambiente de alto nível para a execução do programa,

envolvendo o traçado de gráficos, a apresentação de resultados em janelas, uma estrutura de menus para a seleção das rotinas, funções de seleção de arquivos, função de ajuda ao usuário etc.

Através dos testes iniciais realizados com o programa, podemos concluir que os resultados foram satisfatórios. Os algoritmos de identificação implementados convergiram para os modelos dos sistemas estudados mesmo quando os sinais de entrada e saída apresentavam uma considerável degradação devido à presença de altos níveis de ruído. Aqui é importante mencionar que, a utilização de ferramentas para o pré-tratamento de sinais, contribuíram muito para a obtenção do bom desempenho dos algoritmos de identificação.

Quanto aos demais módulos do programa, responsáveis pela execução das etapas do procedimento de identificação, constatou-se que também apresentaram um desempenho satisfatório, verificando-se as facilidades obtidas com a integração de várias ferramentas num único pacote computacional, em função do número de interações que podem ser necessárias neste procedimento.

Em relação ao ambiente desenvolvido para dar suporte à execução das diversas rotinas do programa, pode-se afirmar, apesar de consideradas as suas limitações, que a sua utilização torna-se bastante simplificada, em função da estrutura de arquivos virtuais, que garantem um processamento mais rápido, e das informações fornecidas ao usuário quando da entrada de parâmetros.

Com a implementação desta primeira versão do programa, observou-se que vários aspectos podem ser melhorados, não somente em relação ao procedimento de identificação, mas também no que se refere à sua interface e ferramentas auxiliares, objetivando-se a obtenção de um pacote mais comercial, no sentido de poder ser melhor aproveitado no ensino em universidades e na aplicação em sistemas reais em indústrias. Desta forma,

como perspectiva de continuidade de estudos e pesquisas na área, propõe-se:

- Em relação à base de métodos de identificação, a inclusão de novos algoritmos onde poderão se consideradas outras abordagens para a identificação, como por exemplo, envolvendo a análise da resposta em frequência dos sistemas;

- Inclusão de novas ferramentas para o auxílio à determinação da estrutura do modelo que melhor represente o sistema;

- Ampliação do módulo de análise e tratamento de sinais, considerando-se a inclusão de novos filtros, com características de resposta tipo passa-alta, passa-faixa, e rejeita-faixa. Além disto, devem ser incluídas no ambiente do programa, maiores informações de auxílio ao usuário em relação a análise do espectro em frequência dos sinais para a especificação dos referidos filtros;

- Em relação ao ambiente do programa, podem ser considerados novos casos de tratamento de erros, como por exemplo, a verificação do espaço em discos de memória permanente, antes do salvamento de arquivos. Além disto, para facilitar a operação do programa devem ser incluídas ferramentas para o gerenciamento de diretórios do sistema operacional, impressão de resultados diretamente do programa, apresentação de gráficos de vários sinais simultaneamente, entre outros.

- Da mesma forma, a especificação do módulo de realização do experimento de aquisição de dados, pode ser revista, incluindo-se uma estrutura de "drivers", onde as características da placa de aquisição de dados acoplada ao computador são armazenadas em arquivos, permitindo-se uma maior flexibilidade de utilização do programa em ambientes industriais. Neste caso, o usuário que possuir um "hardware" de aquisição de dados diferente daquele para o qual o módulo foi desenvolvido, pode

adaptá-lo a este, com a criação de um arquivo que comunica ao programa, as interfaces do computador a serem utilizadas.

- Por fim, dentro de um escopo mais abrangente, prevê-se a inclusão no programa de ferramentas e algoritmos para a identificação de sistemas multivariáveis, considerando-se ainda, a reutilização dos algoritmos implementados no desenvolvimento de pacotes dedicados a casos específicos.

BIBLIOGRAFIA

- [1] Åström K. J., Eykhoff P. "System Identification - A Survey". Automatica, Vol. 7:123-162 (1971).
- [2] Åström K. J. "Maximum Likelihood and Prediction Error Methods". Automatica, Vol. 16:551-574 (1980).
- [3] Bergland G. D. "A guided tour of the fast Fourier transform". IEEE Spectrum, Vol 6:41-52 (1969).
- [4] Cardenas, A. A. "Pacote para Identificação de Processos Biológicos". Tese de Mestrado, UFSC (1988).
- [5] Cardoso Filho M. "Identificação em tempo real de sistemas lineares pelo método da correlação". Tese de Mestrado, UFSC (1979).
- [6] Chen C. T. "Introduction to Linear System Theory". Holt, Rinehart and Winston (1970).
- [7] Daryanani G. "Principles of Active Network Synthesis and Design". John Wiley & Sons (1976).
- [8] Dorn W. S., McCracken D. D. "Cálculo Numérico com Estudos de Casos em Fortran IV". Editora Campus (USP) (1978).
- [9] Eykhoff P. "System Identification. Parameter and State Estimation". John Wiley and Sons, London (1974).

- [10] Fasol K. H., Jorgl H. P. "Principles of model building and Identification". Automatica, Vol. 16:505-518 (1980).
- [11] Foulard C., Gentil S. "Commande et Régulation par Calculateur Numérique. De la Théorie aux Application". Editions Eyrolles, Paris (1977).
- [12] Gentil S. "Etude comparative de diverses méthodes statistiques d'identification de systèmes dynamiques". Tese de doutorado, Université Scientifique Médicale de Grenoble (1972).
- [13] Gertler J., Bányasz C. "A Recursive (On-Line) Maximum Likelihood Identification Method". IEEE Transactions on Automatic Control, Vol. AC-19 (1974).
- [14] Godfrey K.R. "Correlation Methods". Automatica Vol. 16:527-534, (1980).
- [15] Goodwin, G.C., Payne, R.L. "Dynamic System Identification: Experiment Design and Data Analysis". Academic Press, (1977).
- [16] Graupe, D. "Identification of Systems". Robert E. Krieger Publishing Company, (1976).
- [17] Isermann R., Baur U., Bamberger W., Kneppo P., Sebert H. "Comparison of six on line identification and parameter estimation methods". Automatica Vol. 10:81-103 (1974).
- [18] Isermann R. "Practical Aspects of Process Identification". Automatica Vol. 16:575-587 (1980).

- [19] Lima A. C., Alonso C. E. U., Barreto M. R. P. "Ambiente para desenvolvimento e aplicação de algoritmos de Identificação de Sistemas". Anais do 7º Congresso Brasileiro de Automática, pg 614-618, (1988).
- [20] Luz, L.T.O. "Sistema de Identificação, Modelagem, Análise e Projeto de Sistemas de Controle para Sistemas Elétricos de Potência". Tese de Mestrado, UFSC (1990).
- [21] Martinez, M. "Identification de la réponse impulsionnelle d'un système par intercorrélacion ou moyen de séquences binaires pseudo-aléatoires". Tese de doutorado, Université Scientifique Médicale de Grenoble, 1970.
- [22] Menendez, C. "Sur une application de l'identification par filtrage non linéaire à la régulation adaptative en temp réel". Tese de Doutorado, Rennes (1979).
- [23] Moler, C., Liule, j., Bangert, S., Kleiman, S. "PC-MATLAB User's Guide". The MathWorks, Inc, (1986).
- [24] Oppenheim A. V., Schafer R. W. "Digital Signal Processing". Prentice-Hall, (1975).
- [25] Oppenheim A. V., Willsky A. S. "Signal and Systems". Prentice-Hall, (1983).
- [26] Rabiner, L. R., Schafer R.W. "Digital Processing of Speech Signals". Prentice-Hall, (1978).
- [27] Rake, H. "Step Response and Frequency Response Methods". Automatica Vol 16:519-526, (1980).

- [28] Sage, A.P., Melsa, J.L. "System Identification". Academic Press, (1971).
- [29] Saridis, G. N. "Comparision of six on line identification algorithms". Automatica, Vol. 10:69-79, (1974).
- [30] Sinha, N.K., Kuszta, B. "Modeling and Identification of Dynamic Systems". Van Nostrand Reinhold Company, (1983).
- [31] Soderstrom, T. "Convergence Properties of the Generalised Least Squares Identification Method". Automatica Vol. 10:617-626, (1974).
- [32] Soderstrom T., Ljung L., Gustavsson L. "Theoretical Analysis of Recursive Identification Methods". Automatica, Vol. 14:231-244, (1978).
- [33] Strejc V. "Least Squares Parameter Estimation". Automatica, Vol. 16:535-550, (1980).
- [34] Zadeh, L. A. "From circuit theory to system theory". Proc. IRE 50, 856-865 (1962).

A P Ê N D I C E

ISAC

Identificacao de Sistemas Assistida Por Computador

L C M I - U F S C

MANUAL DO USUÁRIO

1 - INTRODUÇÃO

Neste manual, será descrito o pacote computacional ISAC - Identificação de Sistemas Assistida por Computador, destacando-se, dentro de uma abordagem prática, as funções implementadas e o ambiente desenvolvido para dar suporte à sua execução.

2 - ESCOPO DE ABRANGÊNCIA DO PROGRAMA

O ISAC está voltado à identificação de sistemas lineares e monovariáveis, representados por modelos discretos baseados em equações de diferenças, sendo também utilizados, como forma de entrada de dados ou como resultados de alguns métodos de identificação, modelos contínuos dados na forma de funções de transferência.

Como característica principal o programa dispõe de uma base de métodos de identificação, sendo também implementadas rotinas para a geração de sinais, análise e tratamento de sinais e validação de modelos. Além disto, considerando a utilização do programa em ambiente acadêmico, foi implementado um módulo para a simulação de sistemas.

3 - INSTALAÇÃO DO PROGRAMA

O ISAC foi desenvolvido na linguagem MS-Pascal, resultando num código fonte de aproximadamente 13000 linhas. O programa é executado a partir do arquivo "ISAC.EXE" que ocupa 422 kbytes de memória. Além disto, durante a sua execução, o programa pode criar arquivos temporários no diretório corrente. Desta forma, a instalação do ISAC requer como suporte um microcomputador tipo IBM PC-XT que apresente dispositivos de memória permanente de no mínimo 720 kbytes.

O ISAC foi desenvolvido, considerando a sua execução em "overlay". Desta forma, o arquivo contendo o programa (ISAC.EXE) não pode ter seu nome

alterado, e deve permanecer durante a sua execução no diretório em que foi inicializado.

4 - O AMBIENTE DE EXECUÇÃO DO ISAC

No desenvolvimento do programa todas as funções foram implementadas de forma a operarem de maneira independente, sendo desenvolvido um ambiente de alto nível para o gerenciamento destas funções e a realização da interface com o usuário.

Para viabilizar a execução independente de suas rotinas, o programa dispõe de um "diretório de arquivos virtuais", que é responsável pelo armazenamento dos principais tipos de dados utilizados e resultantes de seus diversos algoritmos.

O ambiente de execução do ISAC, compreende um conjunto de funções que permitem o gerenciamento da estrutura de dados do programa, o controle dos algoritmos, através de menus, a entrada de dados e a apresentação de resultados.

4.1 O DIRETÓRIO DE ARQUIVOS VIRTUAIS

A principal finalidade do diretório de arquivos virtuais é a de permitir um rápido acesso aos dados, dispensando-se as operações de interface com dispositivos de memória permanente durante a execução das rotinas do programa.

O diretório de arquivos virtuais envolve os sinais de entrada e saída e modelos de sistemas, que ficam armazenados na área de dados do programa. Os arquivos de modelos têm a extensão .MOD os arquivos de sinais com dados de entrada .ENT e, de saída .SAI. O programa permite a criação de até 20 arquivos de sinais de entrada e saída e 15 arquivos de modelos.

O ISAC trabalha com modelos discretos e contínuos, de até 10ª ordem, descritos, respectivamente, através das equações (1) e (2), sendo utilizada a mesma estrutura de arquivos para os dois tipos de modelos.

$$G(s) = \frac{(b_0 + b_1s + b_2s^2 + \dots + b_ns^n)e^{-as}}{a_0 + a_1s + a_2s^2 + \dots + a_ns^n} \quad (1)$$

$$G(z) = \frac{(b_0 + b_1z^{-1} + b_2z^{-2} + \dots + b_nz^{-n})z^{-a}}{1 + a_1z^{-1} + a_2z^{-2} + \dots + a_nz^{-n}} \quad (2)$$

Os arquivos de sinais podem ter até 10000 pontos. Quando a área de dados é insuficiente para o armazenamento de todos arquivos de sinais, o programa, de forma transparente ao usuário, cria arquivos temporários no diretório corrente, carregando-os automaticamente quando estes forem selecionados.

4.2 - INTERFACE DE ARQUIVOS

É constituída por um conjunto de funções que permitem a visualização do diretório de arquivos e a realização da interface entre a área de dados do programa e a memória de massa.

O acionamento destas funções pode ser realizado em qualquer menu do programa, com as seguintes teclas:

F2 - Diretório: Apresenta o diretório dos arquivos residentes na memória do programa. O uso da tecla F6, permite a visualização dos arquivos de extensão .ENT, .SAI e .MOD armazenados no diretório corrente.

F3 - Carrega: Permite a inserção no programa de arquivos armazenados em memória de massa, sendo que somente podem ser carregados arquivos com nomes diferentes daqueles já existentes no seu diretório.

F4 - Salva: Salva o contexto de um arquivo do sistema em memória de massa, sendo questionada a intenção de se apagar ou não este arquivo do diretório do programa.

F5 - Apaga: Descarta do diretório virtual arquivos considerados desnecessários.

Todas estas rotinas são desativadas com a tecla ESC.

O armazenamento dos dados do programa em memória de massa é realizado em arquivos de dados reais (FILE of Real), sendo os dados escritos e lidos seqüencialmente com comandos do tipo "write" e "read". A estrutura dos arquivos e a seqüência em que são escritos os dados é a seguinte:

- a)arquivos de sinais : número de pontos;
período de amostragem;
dados do sinal;
- b)arquivos de modelos: ordem do modelo;
período de amostragem;
atraso do modelo;
coeficientes a_0, a_1, \dots, a_n ;
coeficientes b_0, b_1, \dots, b_n ,

sendo que se o arquivo for de um modelo contínuo, no campo referente ao período de amostragem, deve ser colocado o valor de controle -999.

4.3 - ESTRUTURA DE MENUS

O controle das atividade do programa é realizado por uma estrutura de menus, onde as funções são selecionadas posicionando-se o cursor sobre o campo desejado e pressionando-se a tecla "enter", ou através das teclas ressaltadas em maiúsculo.

O programa apresenta basicamente 4 menus, estruturados em dois níveis, sendo o primeiro responsável pela seleção dos seus principais módulos e pelo acionamento das funções de interface. Os outros três menus

controlam a seleção das atividades dos módulos de geração de sinais, de simulação e de identificação. Além disto, também são criadas janelas temporárias para a seleção de algoritmos, tipos de modelos e dispositivos de saída. A estrutura de menus do programa e o encadeamento entre estes pode ser visualizada na Fig. 1.

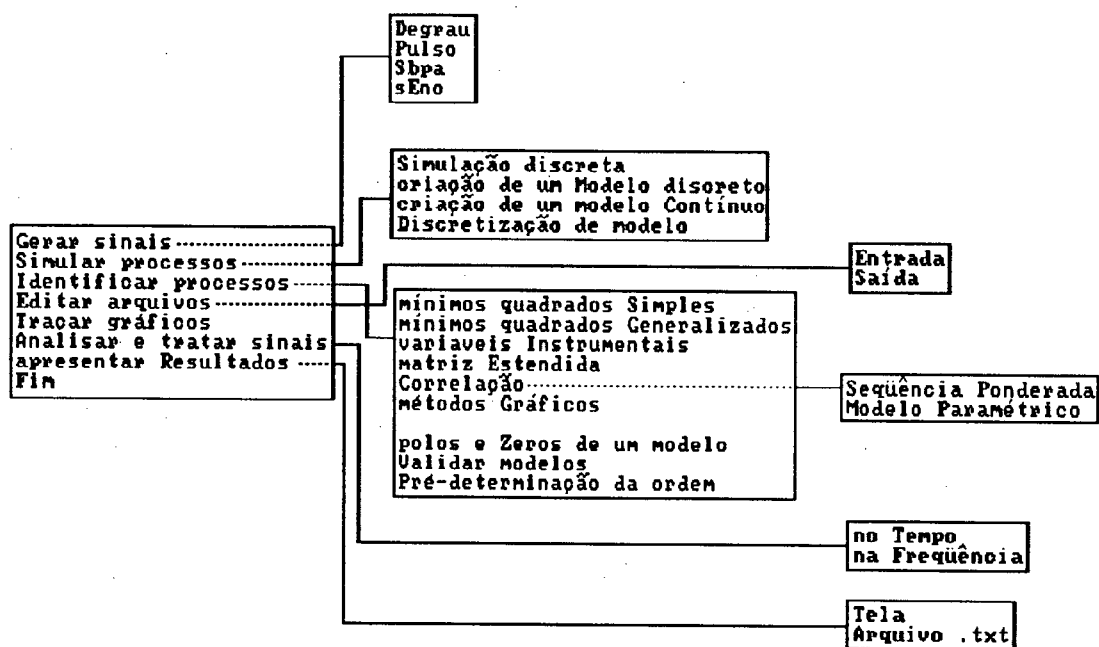


Fig. 1 - Estrutura dos menus do programa ISAC

4.4 - EDIÇÃO DE ARQUIVOS

Para a inserção de sinais no programa, no caso em que estes tenham sido obtidos em experimentos onde a leitura de dados se realiza manualmente, foi desenvolvido um algoritmo que permite a criação e edição de arquivos através do teclado. A utilização deste algoritmo envolve os seguintes aspectos:

- uma janela temporária permite a seleção do tipo de arquivo de sinal, se de entrada ou de saída;

- é lido o nome do arquivo, o número de pontos a serem editados e o período de amostragem do sinal;
- os dados são inseridos seqüencialmente, permitindo-se a inserção (tecla I) de dados entre os já editados e o cancelamento (tecla D) daqueles digitados erroneamente.

4.5 - TRAÇADO DE GRÁFICOS

Traça o gráfico dos arquivos de sinais, selecionados em uma janela com o diretório de todos os arquivos de entrada e saída armazenados no programa. Nesta rotina são disponíveis as seguintes funções:

Z - Zoom: Permite a ampliação de parte do sinal apresentado na tela, sendo para tanto selecionada através de um cursor gráfico a faixa do sinal que se deseja ampliar;

C - Cursor: Coloca o cursor sobre a curva traçada, plotando os valores de seus pontos.

O traçado dos gráficos é realizado em janelas com até 200 pontos, possibilitando-se a visualização de várias páginas de um sinal através das teclas de navegação (setas de posição).

A navegação do cursor nestas duas funções é realizada pelas setas de direção, sendo que para a função ZOOM este é deslocado de 5 pontos, enquanto que para a função cursor o deslocamento é unitário, podendo ser ampliado com a utilização simultânea da tecla "control".

4.6 - APRESENTAÇÃO DE RESULTADOS

Para a visualização do conteúdo dos arquivos residentes no diretório virtual, foram implementadas rotinas de apresentação de resultados, tendo-se como opção a sua impressão em janelas dentro do ambiente ou em arquivos textos a serem criados no diretório corrente. Para tanto, após a

seleção do arquivo a ser apresentado, o programa permite a escolha do dispositivo de saída. Os arquivos de modelos são apresentados em uma única tela do programa, enquanto que, os arquivos de sinais, quando possuírem mais de 20 pontos, são listados utilizando-se diversas janelas, sendo estas selecionadas através das teclas de navegação.

4.7 - AUXÍLIO AO USUÁRIO

Todas as interfaces do programa foram implementadas de forma a auxiliar o usuário na utilização dos seus algoritmos, sendo, por exemplo, listados os valores máximos permitidos durante a leitura de um parâmetro. Além disto, em todas os menus, pode-se acionar a função "ajuda", através da tecla F1, que permite a visualização de informações adicionais sobre o algoritmo a ser utilizado, sendo considerada as seguintes opções:

Pg/Dw - Permite a visualização das informações referentes ao algoritmo seguinte dentro da estrutura do menu em que foi acionada a função;

Pg/Up - Permite a visualização das informações referentes ao algoritmo anterior dentro da estrutura do menu em que foi acionada a função;

I - Apresenta o índice da função, o qual obedece a mesma estrutura descrita para os menus do programa;

ESC - Desativa a janela da função ajuda.

5 - GERAÇÃO DE SINAIS

O módulo de geração permite a criação de arquivos com sinais do tipo degrau, pulso, seqüência binária pseudo-aleatória e sinais compostos por até oito senóides, com até 10000 pontos, sendo requeridos os seguintes parâmetros:

Degrau: - nome do arquivo;
 - número de pontos;
 - amplitude do sinal.

- Pulso:**
- nome do arquivo;
 - número de pontos;
 - amplitude do sinal.
- SBPA:**
- comprimento máximo da seqüência (selecionado pela barra de espaços);
 - nome do arquivo;
 - número de pontos;
 - amplitude do sinal;
- SENO:**
- número de componentes senoidais; (no máximo 8)
 - nome do arquivo;
 - número de pontos;
 - frequência de amostragem (em Hz),
- para cada componente senoidal:
- amplitude;
 - frequência do seno (em Hz);
 - fase (em graus);

6 - Simulação de processos

Simula a realização do experimento de obtenção dos dados de saída do sistema, considerando a inserção de ruídos brancos ou coloridos. Neste módulo são implementadas rotinas para a criação de arquivos de modelos, discretos e contínuos, e para a discretização destes últimos, uma vez que o procedimento de simulação é realizado através de equações de recorrência. Os parâmetros a serem fornecidos em cada um destes algoritmos é o seguinte:

6.1 - Simulação Discreta

- nome do arquivo do modelo discreto a ser simulado;
- nome do arquivo com o sinal de entrada a ser utilizado na excitação do sistema;
- nome do arquivo a ser criado com o sinal de saída resultante da simulação;
- número de pontos a serem simulados;

- para a inserção de ruídos sobre o sinal simulado deve ser fornecido o valor percentual deste sobre o valor médio do sinal;
- escolha do tipo de ruído a ser aplicado: branco ou colorido.

O programa permite a inserção de ruídos brancos e coloridos, sendo estes últimos obtidos com a filtragem dos primeiros, através da equação de recorrência dada pela equação 3.

$$n(k) = d_0v(k) + d_1v(k-1) + d_2v(k-2) + d_3v(k-3) - c_1n(k-1) - c_2n(k-2) - c_3n(k-3) \quad (3)$$

6.2 CRIAÇÃO DE UM ARQUIVO COM UM MODELO DISCRETO

- nome do arquivo a ser criado;
- ordem do modelo;
- atraso do processo (em períodos de amostragem);
- período de discretização com o qual o modelo foi obtido;
- coeficientes do modelo;

6.3 CRIAÇÃO DE UM ARQUIVO COM UM MODELO CONTÍNUO

- nome do arquivo a ser criado;
- atraso do modelo; fornecido em função do período de amostragem com que se fará a sua discretização com vistas a simulação discreta;
- coeficientes do modelo;

obs: A entrada dos parâmetros do modelo é realizada através de polinômios para o numerador e denominador da função de transferência, possibilitando, em ambos os casos, a edição de no máximo 5 polinômios de até 5ª ordem. Para tanto, devem ser fornecidos o número de polinômios, a ordem e os parâmetros correspondentes a cada um destes.

O algoritmo se encarrega da redução dos polinômios para uma forma geral, e verifica se a ordem do denominador é maior ou igual a do numerador, reiniciando a leitura dos parâmetros caso contrário.

6.4 DISCRETIZAÇÃO DE UM MODELO

- nome do arquivo com o modelo contínuo;
- nome do arquivo, a ser criado, com o modelo discreto;
- período de discretização escolhido para o modelo;

7 - ANÁLISE E TRATAMENTO DE SINAIS

Para o tratamento de sinais a serem utilizados no procedimento de identificação, são consideradas duas abordagens: o tratamento interativo através de técnicas de regressão e a filtragem de dados. O primeiro caso é aplicável quando o sinal de saída do sistema apresentar uma forma conhecida como, por exemplo, a resposta ao pulso ou ao degrau, podendo-se facilmente verificar a presença de ruídos sobre o sinal. Outra alternativa é a análise senoidal dos sinais, através do levantamento do seu espectro e o projeto de filtros a serem aplicados sobre os dados de entrada e saída do sistema. A seleção do tipo de abordagem a ser utilizada é feita em uma janela temporária do programa.

7.1 TRATAMENTO DE SINAIS NO TEMPO

Procedimento que permite a modificação das características de um sinal através da substituição, de regiões deste, por retas ou curvas de segunda e terceira ordem que melhor representam os dados selecionados. Além disto, é considerado o tratamento de pontos espúrios, sendo estes substituídos pela média dos seus pontos laterais.

Todo o procedimento é realizado interativamente com a participação direta do usuário na seleção das regiões do sinal a serem substituídos e na escolha do tipo de curva que deve aproximar o comportamento dos dados selecionados.

A seqüência de execução do algoritmo envolve as seguintes etapas e aspectos:

- leitura do nome do arquivo com o sinal a ser tratado;
- leitura do nome do arquivo criado para o armazenamento dos dados resultantes do tratamento de sinais;
- é traçado o gráfico do sinal, sendo permitida a ampliação de parte deste, através da função "ZOOM", de forma a garantir uma melhor visualização da região que se pretende tratar.
- são disponíveis as seguintes opções de tratamento:

- F1 - Substituição de um ponto;
- F2 - Linearização;
- F3 - Regressão Polinomial de 2ª ordem;
- F4 - Regressão Polinomial de 3ª ordem;
- F5 - Recupera a curva anterior;

- a seleção da região do sinal a ser tratada é feita através do posicionamento do cursor, com as setas de direção, e pressionando-se a tecla "enter". Além disto, pode-se utilizar a tecla "END" para que o tratamento se estenda do primeiro ponto selecionado até o final do arquivo.
- ao final do procedimento, é confirmada a criação ou o descarte do arquivo criado, em função dos resultados obtidos.

7.2 TRATAMENTO DE SINAIS NA FREQUÊNCIA

O tratamento de sinais na frequência envolve a análise do espectro em frequência dos dados de entrada e saída do sistema, em função dos quais são fornecidos os parâmetros para o projeto do filtro passa-baixas a ser aplicado sobre ambos os sinais.

A seqüência de execução do algoritmo envolve as seguintes etapas e aspectos:

- leitura do nome dos arquivos de entrada e saída a serem tratados;
- leitura do nome dos arquivos de entrada e saída a serem criados para o armazenamento dos sinais filtrados;

- leitura do número de pontos que devem ser utilizados para o cálculo do espectro em frequência dos sinais;
- são traçados os espectros de ambos os sinais, permitindo-se a visualização, quando for o caso, de outras páginas do espectro, através das teclas de navegação;
- pressionando-se a tecla 'F'(Filtro), é traçado um gabarito para a orientação do usuário no fornecimento dos parâmetros para o projeto do filtro.
- São lidos os valores da atenuação na banda de rejeição do filtro, e de suas frequências de corte, sendo que estas últimas devem ser no mínimo duas vezes menores que a frequência de amostragem do sinal
- o cálculo do filtro e dos sinais resultantes é realizado automaticamente, apresentando-se ao final, os espectros em frequência dos sinais filtrados.

8 - IDENTIFICAÇÃO DE SISTEMAS

Para a identificação de sistemas são considerados vários algoritmos, desenvolvidos segundo três abordagens diferentes: o princípio dos mínimos quadrados; a correlação entre os sinais de entrada e saída e as características de resposta ao degrau.

8.1 - ALGORITMOS BASEADOS NOS MÍNIMOS QUADRADOS

Baseados no princípio dos mínimos quadrados foram implementados 4 algoritmos: mínimos quadrados simples, mínimos quadrados generalizados, variáveis instrumentais e método da matriz estendida; sendo que os três últimos consideram diferentes formas para o tratamento dos ruídos presentes nos sinais de entrada e saída do sistema. Entretanto, para o usuário, a sua utilização envolve uma interface comum, descrita a seguir:

- nome do arquivo de modelo, a ser criado para o armazenamento dos resultados obtidos;
- nome dos arquivos contendo os sinais de entrada e saída;

- número de pontos que devem ser utilizados na identificação.

Os demais parâmetros podem ser inseridos através do teclado ou obtidos de um arquivo de modelo, resultante da utilização de outro método. Para tanto, o programa pergunta ao usuário, se este deseja que aos coeficientes do modelo a ser identificado sejam afetados valores iniciais. Em caso afirmativo, é lido o nome do arquivo contendo estes parâmetros, e em caso contrário devem ser acrescentados os seguintes dados àqueles já descritos:

- ordem do modelo;
- atraso do sistema (em função do período de amostragem);
- se o processo apresenta transferência direta ou não.

Este último parâmetro indica se o modelo a ser identificado deve considerar ou não a estimação do coeficiente b_0 , o qual dentro da estrutura da equação 1, indica se o processo apresenta ou não transferência direta.

8.2 MÉTODO DA CORRELAÇÃO

Este algoritmo é desenvolvido em duas etapas distintas selecionadas em uma janela temporária do programa que compreendem a obtenção da seqüência de ponderação do sistema, e do modelo paramétrico propriamente dito.

A seqüência de ponderação corresponde a saída do sistema a um sinal do tipo pulso unitário. Na sua obtenção é utilizado um algoritmo baseado na correlação entre os sinais de entrada e saída do sistema. No programa ISAC, pressupõe-se a utilização da seqüência binária pseudo-aleatória como sinal de excitação, sendo requeridos os seguintes parâmetros para o algoritmo em questão:

- nome do arquivo com a SBPA;
- nome do arquivo com o sinal de saída resultante;
- nome do arquivo a ser criado para o armazenamento da seqüência de ponderação;

- comprimento máximo da sequência (selecionada pela barra de espaços);
- número de pontos a serem utilizados no algoritmo (múltiplo do comprimento máximo da SBPA);

O procedimento para a obtenção do modelo paramétrico, utiliza o princípio dos mínimos quadrados sobre a sequência de ponderação e envolve os seguintes parâmetros:

- nome do arquivo a ser criado para o armazenamento do modelo identificado;
- nome do arquivo contendo a sequência de ponderação;
- ordem estimada para o modelo;
- atraso do sistema;
- número de pontos a serem utilizados no algoritmo.

8.3 MÉTODOS GRÁFICOS

Foram desenvolvidos quatro algoritmos baseados na resposta do sistema ao degrau sendo os modelos identificados contínuos e pré-definidos. Os resultados de sua utilização dependem diretamente de uma correta amostragem da resposta do sistema e da adequação do sistema identificado ao modelo pré-definido pelo método.

Os resultados obtidos da aplicação destes algoritmos não são armazenados na estrutura de arquivos do programa. Desta forma, como parâmetros de entrada somente requeridos os nomes dos arquivos com o degrau de excitação e com o sinal de saída do sistema.

Após o traçado do gráfico o usuário deve utilizar as opções ZOOM ou EXPANDE, de forma a garantir uma melhor visualização da parte transitória da resposta.

Os métodos disponíveis e os modelos identificados em cada um deles são:

F1 - Primeira ordem	$G(s) = \frac{K e^{-at}}{(TS + 1)}$
F2 - Segunda ordem aperiódico	$G(s) = \frac{K e^{-at}}{(S+t1)(S+t2)}$
F3 - Segunda ordem periódico	$G(s) = \frac{k e^{-at}}{T^2 S^2 + 2\xi TS + 1}$
F4 - Alta ordem aperiódico	$G(s) = \frac{k e^{-at}}{(TS + 1)^n}$

Inicialmente, para todos os métodos, são marcados com o cursor sobre o gráfico os pontos que determinam o atraso e o regime permanente do sinal.

No método de primeira ordem e segunda ordem aperiódico, um segundo gráfico é traçado e o usuário deve marcar dois pontos na parte linear da curva para a determinação de sua inclinação.

Para o método de segunda ordem periódico, devem ser marcados um período de oscilação e o primeiro sobre-sinal da curva sobre o valor de regime permanente.

Por fim, no método para modelos de alta ordem deve ser identificado o ponto de inflexão da curva, sobre o qual é traçada uma reta, que o usuário deve posicionar tangente àquela utilizando as teclas de navegação.

9 - VALIDAÇÃO

Neste módulo, foram implementadas rotinas para a avaliação dos modelos obtidos, no que concerne ao seu desempenho na representação da dinâmica do sistema e à sua própria estrutura. Para tanto, é realizada a simulação

dos modelos obtidos e a comparação dos resultados com as saídas reais do sistema. A verificação da estrutura do modelo é realizada através de seus pólos e zeros, sendo que, em caso de coincidência entre os valores obtidos, pode-se concluir que a ordem estimada para este, é superior àquela necessária para a representação do sistema. Além disto, foi implementada uma rotina, baseada na sequência de ponderação do sistema, que permite uma adequada estimação da ordem do modelo, de forma a se inicializar o procedimento de identificação. Os passos envolvidos na execução destas rotinas e os parâmetros por elas requeridos são descritos a seguir.

9.1 - VALIDAÇÃO DOS MODELOS

- são lidos os nomes dos arquivos com o modelo a ser validado, e com os dados de entrada e de saída do sistema;
- o modelo é simulado, com o sinal de entrada, sendo feita uma comparação entre os dados de saída do sistema e os resultantes da simulação do modelo. São apresentados ao usuário, o erro médio e o erro médio quadrático obtidos;
- opcionalmente, é traçado o gráfico das duas curvas para uma visualização do desempenho do modelo, na representação do sistema, em relação ao sinal de excitação utilizado.

9.2 CANCELAMENTO DE PÓLOS E ZEROS

- é lido o nome do arquivo contendo o modelo a ser avaliado;
- são calculados os pólos e zeros do modelo, sendo os resultados apresentados em janela do programa;
- a análise dos resultados é realizada pelo usuário, cabendo a este decidir se o modelo identificado apresenta uma ordem superestimada ou não.

9.3 PRÉ-DETERMINAÇÃO DA ORDEM

- são fornecidos o nome do arquivo contendo a sequência de ponderação do sistema e, até que ordem do modelo se deseja avaliar;
- numa janela do programa são apresentados os valores resultantes do algoritmo para cada ordem e a variação destes valores em relação a ordem imediatamente superior;
- a ordem que apresentar a maior variação entre estes valores deve ser escolhida para se inicializar os algoritmos de identificação.

Os resultados obtidos podem ser influenciados por erros numéricos e dependem da exatidão da sequência de ponderação utilizada. Desta forma, a apreciação dos resultados é realizada pelo usuário e deve considerar a magnitude dos valores que geraram a suposta variação de maior grandeza. Todavia, o algoritmo tem um bom desempenho na estimação de uma ordem inicial para o modelo, sendo que a sua utilização em conjunto com a rotina de cancelamento de pólos e zeros deve minimizar o número de iterações necessária a obtenção do modelo do sistema.

10 - OUTRAS INFORMAÇÕES SOBRE A UTILIZAÇÃO DO ISAC

* Na realização da interface com o usuário, o programa verifica os parâmetros fornecidos para os algoritmos, rejeitando valores inadequados a sua capacidade, como por exemplo, arquivos com mais de 10000 pontos, ou que sejam absurdos dentro do contexto teórico das rotinas implementadas. Da mesma forma, existe um controle sobre o número de arquivos existentes no diretório do programa, sendo que quando este limite é atingido, ficam desabilitados os algoritmos que criam novos arquivos, até que seja liberado um espaço para tanto.

* Para a entrada do nome de um arquivo em um algoritmo foi implementada uma função de seleção, ativada pela tecla F6, que apresenta a lista dos arquivos armazenados no diretório do programa, sendo a

escolha do arquivo desejado realizada através das teclas de navegação e da tecla "enter". Além disto, o programa verifica se o arquivo selecionado atende aos requisitos do algoritmo em questão, informando o motivo da não aceitação de um determinado arquivo.

* O programa não apresenta um controle sobre erros que envolvam a capacidade da memória de massa utilizada. Desta forma, quando da utilização do ISAC, o usuário deve se assegurar que o disco de trabalho corrente tenha espaço para a criação de arquivos pelo programa. Além disto, a introdução de parâmetros ou arquivos inadequados em um algoritmo, pode levar a um "estouro" da capacidade de uma variável (overflow) sendo o programa imediatamente interrompido. Um fato que pode levar a erros deste tipo é a utilização de um algoritmo de identificação com sinais de entrada e saída não correspondentes.